Biped Locomotion and Stability

A Practical Approach

Jesse van den Kieboom

s1410474

March 2009

Master Thesis

Artificial Intelligence

Dept of Artificial Intelligence
University of Groningen, The Netherlands

Internal supervisor:

Prof. Dr. L. R. B. Schomaker (Artificial Intelligence, University of Groningen)

External supervisor:

Prof. A. J. Ijspeert (Ecole Polytechnique Fédérale de Lausanne, Switzerland)

Abstract

Biped locomotion has proven to be a very hard problem to solve. At current date, there exists no humanoid robot that can move as dynamic and robust as humans do. Even though there has been much research and certainly some interesting progress in biped locomotion, there is certainly no single solution that provides robots with the same capabilities as humans. What seems to be especially lacking is a quantitative method for comparing different solutions. In this work a general method for designing a locomotion controller, independent of the type of robot, is presented. Three distinct stages are identified: Designing the nominal gait, finding a proper encoding of the nominal gait and augmenting the nominal gait to provide stabilization. A nominal gait was successfully evolved using Particle Swarm Optimization algorithms. The nominal gait was encoded into a CPG Network which represents the controller as a dynamical system. Furthermore, a general framework of measuring stability, independent of robot and controller is defined. This allows quantitative comparison of controllers, feedback integration and gaits. Within this framework, stability of the nominal gait was compared with integration of feedback into the CPG. A significant improvement of stability under external perturbation was achieved using simple feedback.

Acknowledgements

First of all, I would like to thank my external supervisor prof. Auke J. Ijspeert who has provided me with support and advice throughout the thesis, and the EPFL for providing the opportunity to do my Master thesis there. I would also like to thank my internal supervisor prof. dr. Lambert R. B. Schomaker who made sure I kept a clear heading and has made it possible for me to do my Master thesis abroad.

Contents

Chapter 1

Introduction

Locomotion has been an active research topic in many different research disciplines for many years. Mobile robots are usually realized using wheels which are easy to control, low in maintenance and readily available. The problem with wheeled locomotion is that it only works reliably on flat or easy terrains. Although this is often the case in controlled environments, in the real world this is less often so. For instance, stairs and elevations such as curbs or large doorsteps are difficult obstacles for wheeled robots. Looking at biological "solutions" for locomotion on land, we can see that legs, rather than wheels, are very able to deal with the kind of difficulties our environment imposes on moving around in it. Be it hexapod, quadruped or biped organisms, they all are very dynamic and flexible systems which can take one many different environments and conditions. They are versatile and robust and as such represent very interesting alternatives to wheeled locomotion in robotics.

Although a significant amount of progress in understanding what the underlying mechanisms are for legged locomotion, the question how these mechanisms are used to provide a dynamic, stable gait on varying terrains for bipeds remains largely unanswered. It is exactly this problem of stability which prevents current solutions for biped, robotic locomotion to be used in practical applications. In other words, they are simply not robust, or capable enough. It is therefore not only interesting to better understand biped locomotion in real-world circumstances from a theoretical point of view, but also from a very practical and social one.

Legged locomotion as a research topic is of course not a new field of research in robotics. There exist a number of humanoid robotic platforms from which Honda's ASIMO is proba-

bly the most well known. Others include the HOAP from Fujitsu and more recently Sony's QRIO and the NAO from Aldebaran Robotics. Besides real robotic platforms, much work has been done in simulations, in both 2D as well as in realistic 3D environments. Furthermore, biped locomotion as a research topic is actively, and publicly promoted and utilized by the RoboCup robot soccer games.

With regard to existing platforms, the problem to be solved is often taken to be the problem of control. In humans as well as in vertebrate animals, rhythmic motor patterns in terms of muscle activation are known the be the result of structures better known as Central Pattern Generators (Grillner & Wallen, 1985; Dimitrijevic, Gerasimenko, & Pinter, 1998; Hooper, 2000). This mechanism characterizes itself as a mechanism that is able generate rhythmic patterns (in this case motor patterns) without the need for specific rhythmic input. Although the use of artificial CPGs is probably not the most widely used method of control at current date, they do present a number of interesting properties (such as stable limit cycles) that are very useful for control. This is even more so the case when multiple CPGs are coupled together to compose a network of oscillators. Once such a network exists, manipulating its state and controlling it with higher level inputs can result in successful control methods (Dickinson, Farley, et al., 2000; Aoi & Tsuchiya, 2005; Kimura, Akiyama, & Sakurama, 1999; J. J. Collins & Richmond, 1994). This combined with integration of sensor information can entrain the controller to the robot and the environment, providing stabilizing properties to the controller in an automatic sense (Taga, 1998; Ito, Yuasa, Luo, Ito, & Yanagihara, 1998).

Conventionally, a robot is first designed mechanically and its control is considered usually at a later and independent stage. More recently, focus is shifting towards more and more co-design of the mechanics and control. In this sense, the mechanics inherently solve part of the control problem intrinsically. Passive-Dynamic walkers (S. Collins, Ruina, Tedrake, & Wisse, 2005) can be considered an extreme case of taking advantage of the interaction of the mechanics of the robot and the environment. These are specifically designed to be needing as little energy and active control as possible, while still being able to move (for example walking down a slope). This is a perfect example of how the environment and the mechanics of the body are incorporated into the design of a robot.

Biological *systems* are extremely well "designed" in terms of mechanics and control. There is no need for high level control for dealing with many small disturbances while

performing certain tasks involving movement. Muscle systems are perfectly capable to deal with these perturbations by themselves and suddenly active, or conscious control becomes much easier. The fact is that our natural environment puts any system operating in it on a high demand of flexibility and robustness, and artificial systems need to be able to handle small disturbances and uncertainty in general (Taga, Yamaguchi, & Shimizu, 1991; Buchli, Righetti, & Ijspeert, 2005; Buchli, Iida, & Ijspeert, 2006; Kimura & Fukuoka, 2004).

It should be clear that if we want the same abilities as natural systems for our artificial systems, they should be inspired on the same principles. Some of these principles include modulation of generated motor signals depending on either intrinsic or extrinsic sensory information, the ability to control the dynamic range of intrinsic sensors (γ motor neurons), reflexes, anticipatory muscle activation and the way sensory information is integrated in all levels of motor control in both a feed forward and a feed backward fashion (Riemann & Lephart, 2002b, 2002a; Thach, Goodkin, & Keating, 1992; J. Collins & Luca, 1993; Mochizuki, Sibley, Esposito, Camilleri, & McIlroy, 2008).

1.1 Overview

One of the issues with current research on biped locomotion for humanoid robots is that there exist no quantitative method for measuring and comparing the quality of a certain system. It is therefore very hard to say whether a certain solution is performing better or worse than existing other solutions. Results of existing research often states that a certain solution works for a certain set of conditions or a certain task, but this a qualitative measurement. Another issue is the lack of a general method for developing a controller for biped locomotion. Such a method should be robot independent and therefore to some extend automatic.

The purpose of this work is twofold. First, a general approach for designing controllers for biped locomotion in a robot independent way is described. The following three distinct stages in designing such a controller are identified and discussed in this order:

1. Designing the nominal gait
2. Finding a proper encoding of the nominal gait

3. Augmenting the nominal gait to provide stabilization

Second, a general framework for evaluating biped locomotion in a quantitative and robot, controller independent manner is developed. This allows for objective comparisons between different solutions independent of the type of robot or controller.

1.2 Environment

All work described in this thesis is being done using the Webots (Michel, 2004) simulation and a model of Fujitsu's HOAP-2 robot. The HOAP-2 (see figure 1.1) is a 25 DOF, rigid humanoid robot which was accurately modeled by Pascal Cominoli for the Webots simulation environment. The choice for this particular environment and robot is based on the experience and availability of real HOAP-2 robot and an accurate model of the robot in simulation at BIRG.

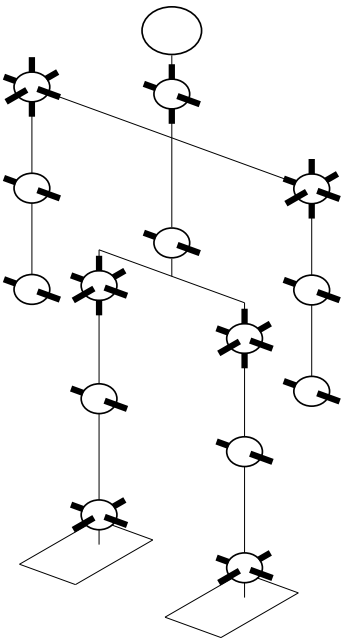

Figure 1.1: Schematic view of the 25 DOF HOAP-2 humanoid robot

Chapter 2

Evolving a Nominal Gait

There are several ways to design a stable nominal bipedal gait. Methods include solving the equations of motion under certain constraints (Zero Moment Point, keeping the foot parallel to the ground), inverse kinematics or manual design by trial and error. These methods either require perfect knowledge and behavior of the model and environment or can require much manual work. Apart from this, solutions are often very model specific and not easily transfered to different robots.

This work takes a different approach than most conventional research in biped locomotion. Instead of designing a specific model of equations to fit the robot, a multi body, professional simulation software is used in which any robot can be easily modeled. Furthermore, an optimization algorithm is used to search for stable open-loop controllers. Combined, this results in a very convenient, practical and easy to use (and extend) framework for developing nominal gaits. The following sections will explain the used optimization algorithm, design of the robot controller and resulting gait.

2.1 Particle Swarm Optimization

The optimization algorithm used to evolve and optimize the locomotion controller is PSO (Particle Swarm Optimization) (Kennedy & Eberhart, 1995). Although the use of Genetic Algorithms is more commonly used for these kind of optimization problems, the PSO algorithm has a very simple and attractive design. It has very few parameters, performs a more directed search than GA and has been shown to converge to (sub)optimal solutions over a wide range of problems and parameter settings. It is also better suited to deal with

continuous values than most conventional Genetic Algorithms.

The algorithm is based on the analogy of particles moving in a N-dimensional parameter space. The location of a particle is given as the parameter vector for that particle, the same way as a gene represents the parameter values for an individual in Genetic Algorithms. The difference is that in PSO, particles/individuals do not generate new generations of particles, but each particle is evolved individually during the optimization process. Apart from a position vector, each particle has a velocity vector whith which it moves in the N-dimensional parameter space (it defines the search direction of the particle). At each "timestep" the particles' *position* is updated using its current velocity. The *velocity*, or search direction, of the particle is updated using the following equation:

$$\begin{aligned} v_{t+1} &= \omega \cdot v_t + c_1 \cdot r_1 \cdot (p_1 - x_t) + c_2 \cdot r_2 \cdot (p_2 - x_t) \\ \dot{x_t} &= v_t \end{aligned} \tag{2.1}$$

where x_t is the position vector at time t, v_t is the velocity vector at time t, ω is a weight called *inertia*, c_1 and c_2 are two positive constants called the cognitive and social factor respectively, r_1 and r_2 two random values in the range $[0, 1]$ and p_1, p_2 are positions of respectively the local best and global best solutions so far.

The inertia weight ω is used to implement a form of friction that can "slow down" the particle and influence the exploration versus convergence properties of the process. Although in its original form, the inertia weight is a constant, the inertia used here is a function of time (ω_t) to allow for larger exploration in the first stages of the optimization, and better convergence properties in the later stages. The update rule for the velocity can be described as searching towards the current best solution of the particle and the global best solution of all the particles. The cognitive and social factors are used to weight the importance of local and global solutions and control to some degree the diversity of the group of particles.

As a consequence of the PSO algorithm, each particle will always keep its best solution in memory and ensures to converge to at least this solution if it does not find any better solution. This differs from Genetic Algorithms where a possible good solution can be lost due to mutation and as such can take longer to converge to good solutions than PSO.

2.2 Controller

Under normal circumstances, the joint angle patterns of a nominal human gait are shown in figure 2.1.

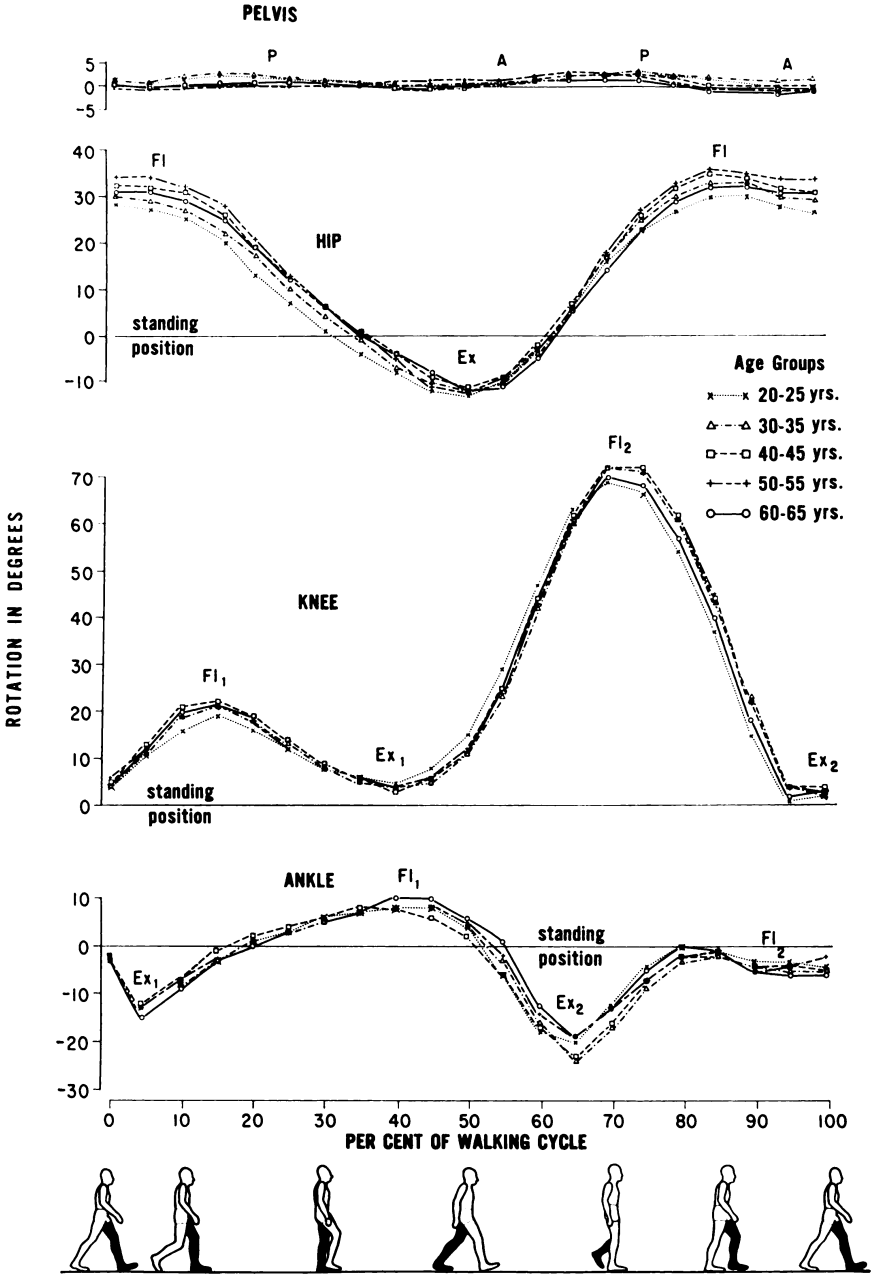

Figure 2.1: Nominal gait patterns (Murray et al., 1964, 1984)

To simplify the problem of finding an optimal solution, the evolved controller uses a single sine wave for each actuated joint to control the joint angle of that joint by position control. Even though actual human joint angle patterns are more complex than a single

sine, it has been shown in previous research that biped controllers can be successfully designed using simple sines for similar robots (Morimoto et al., 2006). Another important consideration that was taken into account in choosing the drive signal is the minimization of search space. Since viable open-loop biped gaits are very likely to cover only a very small part of the parameter space and the number of different solutions are likely to be very small also, it is important to minimize the parameter space as much as possible.

Thus, for each actuated joint the following equation is used to drive the joint angle:

$$\alpha_i = A_i \cdot \sin\left(t \cdot \omega + \phi_i\right) + c_i \tag{2.2}$$

where α_i is the joint angle of the ith joint, A_i is the amplitude of the ith joint, ω is the frequency of the oscillator, which is the same for all joints, ϕ_i is the phase of the sine and c_i is a constant offset.

The joints evolved in this experiments are the frontal and lateral hip joints, the knee joint and the frontal and lateral ankle joints (for reference, see figure 1.1). The movement of the lateral ankle joint is fixed to be the same as the contra-lateral hip joint (e.g. the left lateral ankle joint has the same sine wave pattern as the right lateral hip joint) and the legs are fixed in anti-phase. Also, the phase ϕ_i and offset c_i of the lateral hip joints (and consequently also for the lateral ankle joints) are fixed at respectively $\frac{\pi}{2}$ and 0. The fixed phase ensures that the sequence starts with the load of the robot on one leg to create a situation where the locomotion can be "safely" initiated. The fixed offset is set since the gait should be laterally symmetric and it would not make sense to allow an offset other than 0. The offset of the frontal hip joint is also fixed to 0 since at the start the robot is in a standing position. For the other joints (knee and frontal ankle), all three parameters are evolved.

A last parameter specifies a period of time at the beginning of the locomotion to linearly scale all the joint positions to their respective start positions as given by evaluating the associated sine at $t = 0$. This would allow the robot to have a somewhat smooth transient between the starting upright position and the starting position of all the sine waves. After this startup period, the sine controllers are turned on and periodic motion begins. Lastly, to ensure forward motion the upper body joint is set at a fixed 20° angle shifting the center of mass of the robot a little forward. Table 2.1 summarizes the parameter boundaries for each of the actuated joints. The experiment is run for four different settings of the sine wave frequency (0.50, 0.75, 1.00 and 1.25 Hz). For each of

these frequencies the startup period has an upper boundary set to the respective period $T = \frac{1}{f}$.

Table 2.1: Parameter Boundaries

	A	ϕ	c
Lateral Hip	$[0, 20]$	$-\pi$	0
Frontal Hip	$[0, 60]$	$[-\frac{\pi}{2}, \frac{\pi}{2}]$	0
Knee	$[0, 60]$	$[-\pi, \pi]$	[0, 30]
Frontal Ankle	$[0, 30]$	$[-\pi, \pi]$	[-30, 30]
Body	20		
Frequency	[0.50, 0.75, 1.00, 1.25] Hz		
Startup	[0, T]		

2.3 Experiment

The experimental environment used is the Webots simulator with the HOAP-2 model robot. The initial position of the robot is an upright position at the origin of the webots world. The world consists of the robot and an infinite, flat ground surface. The fitness function used for the PSO algorithm is the euclidean distance from the origin of the world to the position of the robot. The fitness function is very simple and naive, but nevertheless rewards important properties of the gait such as walking straight rather than in a curve and walking speed.

All the experiments are run with 100 particles and 400 iterations. The inertia ω_t of the PSO is set to $1.2 \cdot 0.998^t$ which results in the behavior of the inertia as a function of time as displayed in figure 2.2. This function for the inertia was determined experimentally and ensures strong exploration in the early stages and better convergence in the later stages. It should be noted that for approximately the first 100 iterations, the inertia value is larger than one. This essentially means particles will "overshoot" and this is deliberately chosen as to increase the coverage of the search space. For the behavior at the boundary of the PSO (when particles hit the parameter boundaries) a damped reflective wall is used: $v_{t+1} = -0.6 \cdot v_t$. The cognitive and social factors of the PSO are both set to the

commonly used value of 2.0. The robot is allowed to walk for a maximum of 60 seconds, and the run is preliminary terminated if the robot falls over (marking the distance up until that point as the fitness of the run).

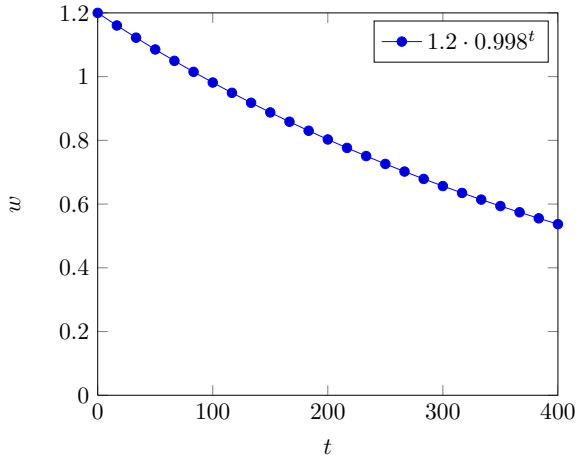

Figure 2.2: The inertia value w as a function of the iteration t

For each of the frequencies, the experiment was repeated 10 times, which results in a total of 40 runs.

2.4 Results

A typical progress of the optimization process can be seen in figure 2.3. It takes a long time before any solution is found that can walk any distance. Furthermore, from the 10 performed runs for each frequency, on average only 2 runs result in a usable gait. As can be seen in figure 2.3, once a particle finds a solution in the right direction, the process is able to converge to a (sub)optimal solution.

2.4.1 Obtained gaits

For each of the four frequencies a comparable gait is found. The fitness value of the best solutions are displayed in table 2.2. A video of the obtained gaits can be found at `http://birg2.epfl.ch/~jvanden/videos/thesis/optimized_gaits.avi`.

The obtained joint angle patterns are shown in figures 2.4 to 2.7. In these figures the solid line indicates the control pattern while the dashed pattern is the actual servo position.

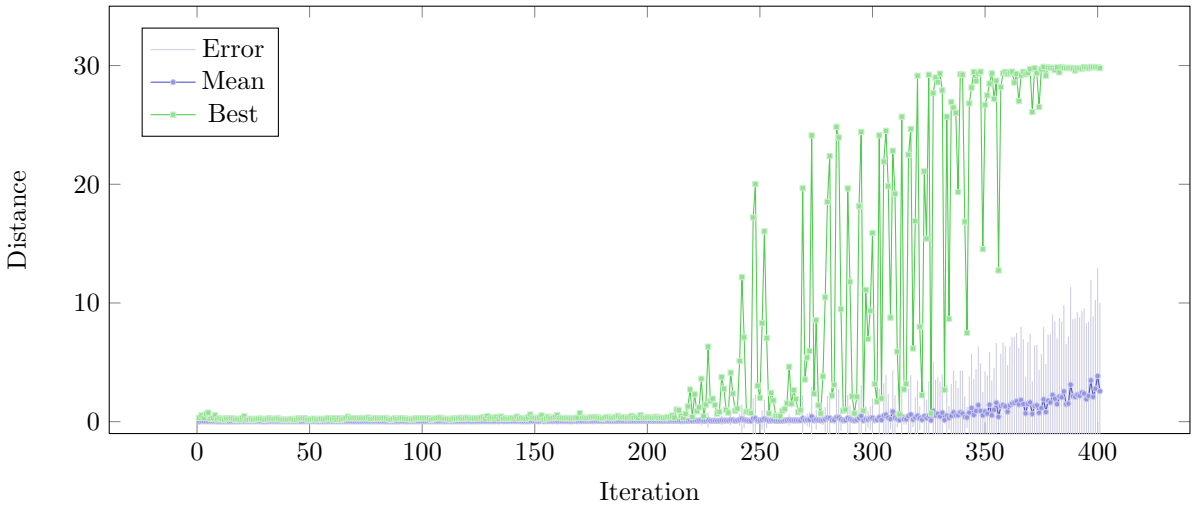

Figure 2.3: Example of optimization process for $f = 0.75$ Hz

Table 2.2: Best solutions

Frequency	**Fitness**	**Speed**
0.50	14.36 m	0.24 m/s
0.75	21.06 m	0.35 m/s
1.00	27.49 m	0.46 m/s
1.25	30.56 m	0.51 m/s

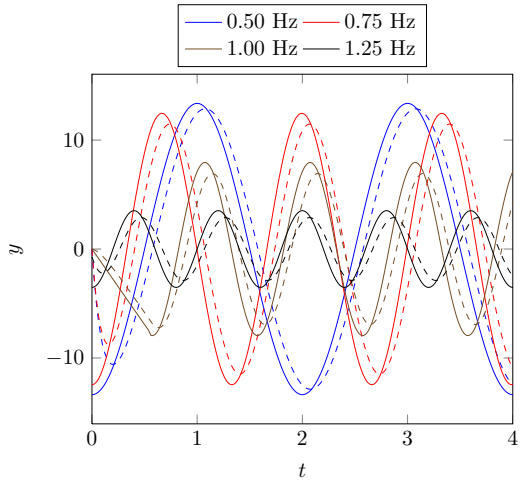

Figure 2.4: Lateral hip trajectories

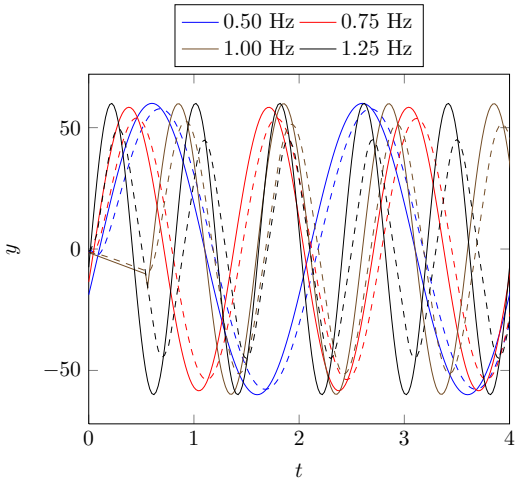

Figure 2.5: Frontal hip trajectories

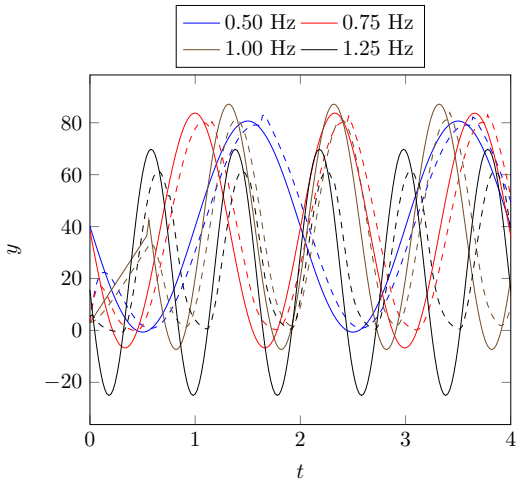

Figure 2.6: Knee trajectories

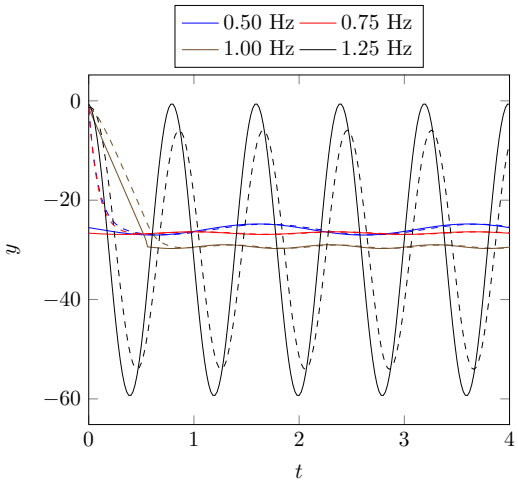

Figure 2.7: Frontal ankle trajectories

2.5 Discussion

The evolved patterns are similar between the different frequencies with the exception of the 1.25 Hz gait which has a very large ankle amplitude with respect to the other gaits. Furthermore, the lateral amplitude of the hip and consequently the ankle joint increases with lower frequencies. This can be explained by the fact that the lower the frequency, the larger the step size and the more lateral compensation is needed to stabilize the gait. It can also be seen that the actual servo position follows the controller pattern closely in general. The exception is the knee joint which cannot be negatively bent, thus resulting in actual servo position differing from the controller pattern. Since the gaits are optimized for distance, they are quite fast, and consequently, higher frequency gaits are faster than the lower frequency gaits.

Due to the fitness function, the obtained gaits are very explosive and very much tuned to the simulation environment. This caused the transfer to a real HOAP-2 robot to be impossible in a straight-forward way.

Even though the solution space for open-loop biped locomotion with simple sine controllers is very small, the PSO algorithm is able to find open-loop stable gaits for each of the frequencies. This can be attributed to the use of top-down knowledge to provide appropriate parameter boundaries to decrease the search space.

Chapter 3

Central Pattern Generators

Several different research groups have taken the Central Pattern Generator approach to implement controllers for biped locomotion especially with regard to stabilization. It has been shown that humans have structures in the spinal cord that enable the generation of rhythmic motor patterns without the need for any rhythmic input. The concept of a artificial Central Pattern Generator is a simplification of such a structure while retaining similar properties and generating rhythmic patterns. The concept of a CPG as a generator of motion patterns is very general and the concept alone does not provide a straightforward method of designing a CPG. Considerations such as the type of oscillator, implementing a nominal locomotion trajectory, incorporating feedback and the structure of the CPG network are all important design decisions. Here a number of different approaches taken by research groups in the field of biped locomotion are reviewed.

3.1 Oscillator Type

The type of oscillator is one of the most important design choices to be made. Probably the most basic and most well defined oscillator is the Hopf oscillator. It is defined as a dynamical system (a system of differential equations) of a perfect sine (when on the limit cycle) in Cartesian space:

$$\begin{aligned} \dot{x} &= \gamma(\mu^2 - r^2)x - \omega y \\ \dot{y} &= \gamma(\mu^2 - r^2)y + \omega x \end{aligned} \tag{3.1}$$

With $r = \sqrt{x^2 + y^2}$, μ the amplitude of the oscillator, γ a parameter which determines the speed of convergence to the amplitude and ω the frequency of oscillation. This oscillator has several advantages. It is well defined in terms of amplitude and frequency which makes it easy to control and it has strong stable limit cycle properties. If therefore additional components like feedback information are introduced in the equations, the oscillator will smoothly incorporate this "perturbation" and once it is gone, converge back to its original configuration. Due to its simplicity, it is also easy to extend to more complex oscillators which can adapt intrinsic frequencies, have different speeds of convergence for both state variables and because they can represent perfect sinusoidal functions, can be coupled to output any periodic function (Righetti & Ijspeert, 2006, 2008).

A disadvantage of this oscillator is that it is not straightforward to design phase coupling of two oscillators without disturbing the amplitude of the oscillator (why this is will be explained in the next chapter). To avoid this problem the oscillator can be expressed in phase space, rather than in Cartesian space. A simple derivation will show that the same oscillator can be defined in phase space as:

$$\begin{aligned} \dot{r} &= \gamma_r(\mu^2 - r^2)r \\ \dot{\phi} &= \omega \end{aligned} \tag{3.2}$$

Since phase ϕ and amplitude r are now fully decoupled, phase coupling of two oscillators can now be easily implemented without affecting the oscillator amplitude. However, the disadvantage of this respresentation is that is not straightforward how to integrate feedback (whereas in Cartesian space it is easy to add terms to x or y).

Two other approaches take inspiration from artificial neural networks to built oscillators to generate walking patterns. The first is the well known Matsuoka oscillator which is defined as (Matsuoka, 1985, 1987):

$$\begin{aligned} \tau \dot{u}_1 &= -u_1 - w y_2 - \beta v_1 + u_0 \\ \tau \dot{u}_2 &= -u_2 - w y_1 - \beta v_2 + u_0 \\ \tau' \dot{v}_1 &= -v_1 + y_1 \\ \tau' \dot{v}_2 &= -v_2 + y_2 \end{aligned} \tag{3.3}$$

$$y_i = f(u_i) = \max(0, u_i), (i = 1, 2) \tag{3.4}$$

where u_i is the inner state (representing the membrane potential) of the ith neuron, y_i is the output of the ith neuron, v_i represents the self-inhibition effect of the ith neuron,

u_0 is an external input, w is a connecting weight and τ, τ' are time constants of inner state and adaptivity respectively (τ specifies the rise time when given the step input and τ' the time-lag of the adaptation effect).

One oscillator is defined by two neurons which through their inhibitory connections alternate in their output (y_1 and y_2) and model flexor and extensor forces. By using $y = y_2 - y_1$, a periodic signal can be produced for which the frequency depends on τ and τ' (proportional to $1/\tau$) and the amplitude depends on u_0.

The model is interesting due to its biological inspiration by using a neuronal model to implement the oscillator. The complexity of the model does however make it difficult to design arbitrary patterns. The shape and stability of the pattern depends heavily on the choice of the open parameters and in extension also makes the coupling of multiple oscillators a non trivial process. Nevertheless, it has been shown (Taga et al., 1991) that it can be applied successfully to biped locomotion where the oscillator is entrained with the mechanics of the biped in an automatic sense. It should be noted that here the biped model was simplified since it could move in the sagittal plane only, was accurately parametrized and used only in numerical simulations.

A comparable, but different neuronal approach is taken recently by (Zaier & Nagashima, 2006; Zaier & Kanda, 2007). In this work a Recurrent Neural Network is used to generate piecewise-linear patterns for locomotion. Here a single neuron i is modeled as:

$$\varepsilon_i \frac{dy_i}{dt} + y_i = \sum_j c_{ij} y_j \tag{3.5}$$

where y_i is the output of neuron i which gets inputs from other neuron's outputs y_j through connections with weights c_{ij}. ε_i is a timing constant of neuron i. This provides an interesting approach to pattern generators as piecewise-linear functions are simple to define and the neuronal model ensures smooth output of the linear functions. Initial parameter settings for ε_i have to be made appropriately for the model to function at all, and have to be tuned by hand. It is unclear what the exact convergence properties of the system are and how feedback can be integrated.

Considering the short review of used oscillators, the Hopf oscillator seems to be the most attractive. Its simplicity and mathematically derived definition makes it easy to understand and analyze. It can be easily controlled through well defined parameters such

as amplitude, frequency and phase, and has a stable, harmonic limit cycle. Therefore, feedback can be incorporated without loss of the intrinsic pattern. Moreover, it has been successfully used to entrain and learn arbitrary rhythmic signals and phase couplings (Righetti & Ijspeert, 2006, 2008; Righetti, Buchli, & Ijspeert, 2006; Buchli, Righetti, & Ijspeert, 2006).

3.2 Oscillator Output

Once an appropriate oscillator is chosen, it has to be defined how the oscillator will drive the robot. Most approaches use either force/torque or position control. A decision has also to be made on how many oscillators drive how many joints. Depending on the complexity of the generated pattern, it might be needed to have multiple oscillators drive a single joint. On the other hand, a single oscillator might also be able to drive multiple joints simultaneously, or a single oscillator might drive the robot by driving a single joint and drive all the other joints using inverse kinematics.

Torque control is often used in numerical simulations where the model and the environment are well defined and constraints for locomotion are given by equations of motion (Taga et al., 1991; Aoi & Tsuchiya, 2005). Although it might be easier or better to implement more compliant systems with torque control, the reality is often that robots are designed with high gear ratio servos and are therefore rigidly controlled by position control instead of providing certain force moments to each of the joints.

3.3 Oscillator Coupling

When considering the symmetry and periodicity of a human gait, the joint angle position patterns, of the joints that are active during locomotion, are strongly phase-coupled. The hip joints that move the leg in the sagittal plane for instance, are moving in anti-phase to provide swing and stance phases alternatively. Similarly, the lateral hip and ankle joints have the exact same phase to keep the body in an upright position. Such phase couplings can be implemented by linking oscillators together. A simple phase locking

coupling scheme for the Phase Oscillator is the following (Morimoto et al., 2006):

$$\begin{aligned} \dot{\phi_i} &= \omega_i + K_i \sin(\phi_j - \phi_i) \\ \dot{\phi_j} &= \omega_2 + K_j \sin(\phi_i - \phi_j) \end{aligned} \tag{3.6}$$

where ω_1 and ω_2 are the frequencies of the oscillators, and the phase difference obtained through the coupling is given by $\phi_j - \phi_i = \frac{\omega_j - \omega_1}{K_i + K_j}$.

The same method can be used for the Hopf oscillator defined in phase space. When defined in Cartesian space, phase coupling behavior for fixed phase relations can be implemented by taking inspiration from the theory of symmetric coupled cells (Righetti & Ijspeert, 2008):

$$\begin{aligned} \dot{x_i} &= \gamma(\mu - r_i^2)x_i - \omega_i y_i \\ \dot{y_i} &= \gamma(\mu - r_i^2)y_i + \omega_i x_i + \sum k_{ij} y_j \end{aligned} \tag{3.7}$$

where k_{ij} is a coupling constant for coupling of oscillator i with oscillator j. It is shown that given different coupling matrices, the same network of oscillators can provide the four basic walking patterns of a quadruped gait. The same methodology can be applied to biped locomotion. Since this approach can only generate very specific phase couplings (such as π or 0.5π), a more generic coupling scheme, using a 2D rotation matrix, can be used to implement any arbitrary phase coupling (see also section 4.2).

The described coupling schemes provide automatic phase coupling between oscillators such that when perturbed they will converge back to the specified phase relation. This is of course a very welcome property because it is ensured that the couplings are well encoded into the network. For the Matsuoka oscillator similar couplings between oscillators can be defined as is shown in (Taga et al., 1991), although it is not explicitly stated how the exact phase relations are obtained. (Aoi & Tsuchiya, 2005) uses a similar approach, but here specific inter oscillators are defined which provide the phase coupling of the different joints.

It is of course also possible to explicitly define oscillator phase differences without coupling the oscillators together (Zaier & Kanda, 2007), but without coupling, additional control structures have to be defined to ensure correct phase differences when dealing with stabilization of the gait.

3.4 Nominal Trajectory

The definition of the nominal trajectory is also very important when designing CPG based locomotion. The nominal trajectory defines for each joint its periodic pattern which provides an approximate stable gait under ideal circumstances. Here inverse kinematics is a commonly used method to derive the nominal trajectory from a certain set of constraints (for example the trajectory of the foot) (Aoi & Tsuchiya, 2005; Taga et al., 1991). This does however require an accurate model of the robot. Other approaches often simply analyze the trajectories for human locomotion and manually tune the nominal trajectory to be approximately correct. Although this method does not provide strong stability properties, the use of feedback control for stabilization and the ability to adapt the locomotion patterns using this feedback, provides the possibility to entrain a stable walking gait. Even if the exact model is given and a derived nominal gait is stable in numerical simulations, in real world environments these gaits also need stabilization control due to imperfections in the robot and perturbations from the environment. It is thus essential to be able to stabilize the nominal trajectories using feedback.

3.5 Feedback

It is well known that stable locomotion is almost impossible with an open-loop controller. It has been shown that when passive dynamic elements are incorporated in the design of the robot and extensive optimization functions are used, open-loop controllers can be designed which provide self stabilizing properties (Mombaur, Bock, & Longman, 2000; Mombaur, Longman, Bock, & Schlöder, 2005). It is however essential that the model is accurate and well known. Furthermore, it is only stable for small perturbations and in real world situations there is often the need for active stabilization strategies. CPGs provide for excellent integration of feedback because they can be easily modulated through their phase, frequency and amplitude parameters. For oscillators which have stable limit cycle properties this is even more so the case, because these oscillators are able to converge back to their original trajectory after perturbation.

One of the most extensively used methods for stabilization is based on the ZMP (zero moment point) (Sugihara, Nakamura, & Inoue, 2002). The ZMP is a point on the ground where the resultant of the ground reaction force acts. As such, stable locomotion can

be achieved when the ZMP stays with in the foot support area of the robot. Although the ZMP criterion is often used in path planning to generate motion trajectories that are stable, the ZMP can also be used in direct feedback control. Deviations of the ZMP towards the edges of the foot support area can be fed back into the CPG.

One important method of stabilization which is easily applied to CPGs is "Phase Resetting". With phase resetting, the interaction between the robot and the environment is entrained in the CPG. The idea here is that one step in a bipedal gait consists of two phases, the stance phase and the swing phase. In order to provide a more stable locomotion, the switch from swing to stance and stance to swing phase is initiated by feedback from sensors in the foot. This way when a leg is in swing phase and load on the sole of the foot is detected, a phase reset ensures that the leg will quickly go to the stance phase to be able to support the load of the body. Phase resetting can be achieved simply by resetting the phases of the oscillators externally using feedback. An interesting alternative is to incorporate the ability to switch between phases into the oscillator themselves as is shown in (Righetti & Ijspeert, 2008).

Other additional methods for stabilization include adjusting to body roll and pitch information (Righetti & Ijspeert, 2006; Zaier & Nagashima, 2006; Aoi & Tsuchiya, 2005) and reflexes such as the stretch reflex (based on reflexes found in vertebrates) (Taga et al., 1991). Although extensive research exists on the role of muscle reflexes in stabilization of human walking (Duysens, Trippel, Horstmann, & Dietz, 1990; Yang & Stein, 1990; Bastiaanse, Duysens, & Dietz, 2000; Zehr & Stein, 1999; Pearson, 1993; Schillings, Wezel, Mulder, & Duysens, 2000) in response to cutaneous and load sensory feedback, these findings, though they serve as a great starting point, are not applied as much to robot bipedal locomotion.

A last interesting approach to stability is the concept of entrainment of the mechanics of the robot and the environment. Through the use of feedback and adjustment of the CPG controller, natural frequencies of the mechanics of the robot can be found and entrained to provide stability (Morimoto et al., 2006; Taga et al., 1991).

The combination of adaptable CPG frameworks and methods to entrain frequencies and phases which are automatically derived from the gait dynamics of the robot shows to be a very promising approach to stabilization of locomotion, especially when designing a generic control framework capable of dealing with different robot morphologies.

In summary, the most common types of feedback are body posture, body acceleration and load measurements on the feet. Body posture information is used to provide stability of body roll and pitch, and forces detected at the feet are used to measure load distribution and phase transitions.

3.6 Discussion

It has been discussed that pattern generators exist in vertebrates and that the motor patterns are modulated by sensory feedback. Using CPGs for a biped locomotion controller provides many interesting and useful properties to encode motor patterns and integrate feedback. The types of feedback used in most of todays solutions for gait stabilization are simple, and although they do provide for a more stable gait, there is much room for improvement. For a generic framework, strict assumptions on the morphology of the robot cannot be essential for stabilization. An autonomous stable locomotion controller should be able to deal with uneven terrain (such as small objects, slopes) and perturbing forces such as wind, or pushes, without the need for higher control strategies. Once such a system exists, higher level strategies could be implement to adjust for larger perturbations or path planning. Inspiration can be taken from the extensive research on reflex systems in biological systems such as humans, which are designed to deal with uncertainties in the environment.

Chapter 4

CPG for Biped Locomotion

The sine controller has shown to be capable of generating joint angle patterns that provide a stable solution to biped locomotion for the HOAP-2 in open-loop. A problem with this controller is that it is very rigid, in the sense that the output driving each of the joints is a direct function of time. Any behavior other than this direct output signal has to be designed, for instance by appropriately warping time, amplitude modulation and frequency modulation. To move away from the direct relation to time, the system can be modeled as a set of differential equations (a dynamical system) and as such is able to provide intrinsic properties such as smooth transitions and easier integration of feedback.

By moving from the sine controller to a CPG based controller, decisions have to be made on the type of oscillator and the structure of the oscillator network. To benefit from the evolved sine controller, there should exist an exact representation of the sine wave patterns in the CPG network so that the same pattern as was evolved before can be used.

4.1 Base Oscillator

The base oscillator should be able to generate a sine wave given a well defined parameterization in terms of amplitude and frequency. It should also allow for any arbitrary phase difference between two or more oscillators (to ensure the correct phase relationships). The Hopf oscillator satisfies the above properties and is used as a basis.

The oscillator can be either represented in Cartesian space, or in phase space. In

Cartesian space, its general form is defined as:

$$\begin{aligned} \dot{x} &= \gamma_x(\mu^2 - r^2)x - \omega y \\ \dot{y} &= \gamma_y(\mu^2 - r^2)y + \omega x \\ r &= \sqrt{x^2 + y^2} \end{aligned} \tag{4.1}$$

With γ_x, γ_y constants which determine speed of convergence to the limit cycle of the oscillator, amplitude μ and frequency ω. Here y (or x by shifting the sine oscillator phase with $\frac{1}{2}\pi$) can be used to drive the joint.

In phase space, the oscillator is defined as:

$$\dot{r} = \gamma_r(\mu^2 - r^2)r \tag{4.2}$$

$$\dot{\phi} = \omega \tag{4.3}$$

Here the constant γ_r determines the speed of convergence to the desired amplitude μ and the frequency is defined as ω. To match the sine controller, the driving signal can be defined as $r \cdot \sin(\phi)$.

To include the desired sine wave offset C in the oscillator, an additional state variable is introduced to the dynamical system:

$$\dot{c} = \gamma_c(C - c) \tag{4.4}$$

In Cartesian space, the oscillator needs to be modified by substituting x by $x - c$. For the oscillator in phase space, c can simply be added to the drive signal resulting in: $r \sin(\omega) + c$. Implementing the offset as a part of the dynamical system has the advantage that it has converges back to the original offset when perturbed and provides smooth transitions if the desired offset is changed.

4.2 CPG Network

Each joint of the robot is driven by an individual Hopf oscillator. These oscillators need to be coupled together to be able to define the correct phase differences needed to generate the correct drive for each of the joints.

4.2.1 Phase Coupling

An arbitrary phase coupling between oscillators (i and j) in Cartesian space can be defined as a rotation matrix on x and y:

$$\dot{x_i}' = \dot{x_i} + \sum_j R_{x_{ij}} \tag{4.5}$$
$$\dot{y_i}' = \dot{y_i} + \sum_j R_{y_{ij}}$$

$$R_{x_{ij}} = x_j \cos(\psi_{ij}) + y_j \sin(\psi_{ij})$$
$$R_{y_{ij}} = -x_j \sin(\psi_{ij}) + y_j \cos(\psi_{ij})$$

This provides an arbitrary phase coupling with a relative desired phase difference between oscillators i, j of ψ_{ij}.

For the phase space representation of the oscillator, the phase coupling is much simpler and more elegant. Since in phase space the amplitude and phase are cleanly separated, the coupling can be defined as:

$$\dot{\phi_i} = \omega_i + \sum_j R_{ij} \tag{4.6}$$
$$R_{ij} = \beta_{ij} \sin(\phi_j - \phi_i - \psi_{ij})$$

With coupling strength β_{ij}, phases ϕ_i, ϕ_j and desired relative phase difference ψ_{ij}.

4.2.2 Improved Phase Coupling in Cartesian Space

Since phase coupling in Cartesian space with a rotation matrix, as defined in equation 4.5, changes the effective amplitude of the oscillator, it is harder to design such oscillators to output a certain amplitude, as well as it makes it hard to translate the sine controller to this type of oscillator. In order to have phase coupling in Cartesian space without influencing the amplitude of the oscillation, a new oscillator is designed starting from the coupled phase space oscillator as defined in equation 4.6.

We start from the definition of the oscillator and its phase coupling in phase space:

$$\begin{aligned} \dot{r}_i &= \gamma_{r_i}(\mu_i^2 - r_i^2)r_i \\ \dot{\phi}_i &= \omega_i + \sum_j \beta_{ij} R_{ij} \\ R_{ij} &= \sin(\phi_j - \phi_i - \psi_{ij}) \\ \dot{c}_i &= \gamma_{c_i}(C - c) \end{aligned} \tag{4.7}$$

To rewrite the phase space equations to Cartesian space we use:

$$\begin{aligned} x &= r\cos\phi + c \\ y &= r\sin\phi \end{aligned} \tag{4.8}$$

$$\begin{aligned} \dot{x} &= \dot{r}\cos\phi + r\cos'\phi + \dot{c} \\ \dot{y} &= \dot{r}\sin\phi + r\sin'\phi \end{aligned} \tag{4.9}$$

Substituting $\dot{r}$ and $\dot{c}$ we get:

$$\begin{aligned} \dot{x} &= \gamma_r(\mu^2 - r^2)r\cos\phi - r\sin\phi\cdot\dot{\phi} + \dot{c} \\ \dot{y} &= \gamma_r(\mu^2 - r^2)r\sin\phi + r\cos\phi\cdot\dot{\phi} \end{aligned} \tag{4.10}$$

$$\begin{aligned} \dot{x} &= \gamma_r(\mu^2 - r^2)(x - c) - y\dot{\phi} + \gamma_c(C - c) \\ \dot{y} &= \gamma_r(\mu^2 - r^2)y + (x - c)\dot{\phi} \end{aligned}$$

Now we rewrite the coupling part R_{ij} of $\dot{\phi}_i$ as defined in equation 4.7 to Cartesian space. Here $x' = x - c$ is used for better readability.

$$\begin{aligned} R_{ij} =& \sin(\phi_j - \phi_i - \psi_{ij}) \\ =& [\sin(\phi_j)\cos(\phi_i) - \cos(\phi_j)\sin(\phi_i)]\cdot\cos(\psi_{ij}) - \\ & [\cos(\phi_j)\cos(\phi_i) + \sin(\phi_j)\sin(\phi_i)]\cdot\sin(\psi_{ij}) \\ =& \left(\frac{y_j}{r_j}\frac{x'_i}{r_i} - \frac{x'_j}{r_j}\frac{y_i}{r_i}\right)\cdot\cos(\psi_{ij}) - \left(\frac{x'_j}{r_j}\frac{x'_i}{r_i} + \frac{y_j}{r_j}\frac{y_i}{r_i}\right)\cdot\sin(\psi_{ij}) \\ =& \frac{1}{r_j r_i}\cdot\left[\left(y_j x'_i - x'_j y_i\right)\cdot\cos\psi_{ij} - \left(x'_j x'_i + y_j y_i\right)\cdot\sin\psi_{ij}\right] \end{aligned} \tag{4.11}$$

This results in the an arbitrary phase coupled oscillator in *Cartesian* space which only influences the phase of the oscillator and not the amplitude. This allows to have both easy

phase coupling (as with the oscillator in phase space), and the advantage of integrating feedback directly on x and/or y.

4.2.3 Network Structure

The network can now be constructed using the oscillators and coupling terms as described before. It is common practice to structure the network to match the physical structure of the robot. Figure 4.1 shows a schematic of the structure of the CPG network used for the HOAP-2. The two legs are coupled at the frontal hip oscillators. Furthermore, there is a direct coupling between the lateral hip and ankle oscillators since they should be strongly coupled to keep the body in an upright posture when one of the oscillators would be perturbed (the other would quickly synchronize).

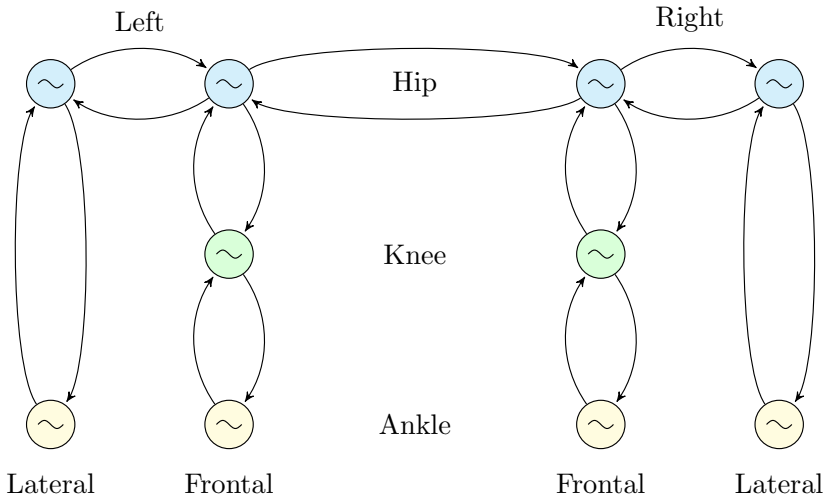

Figure 4.1: CPG network structure

The coupling of the oscillators is arbitrary in the case where a perfect sine is encoded in the system and all oscillators are on the limit cycle (e.g. all oscillators are already phase-locked). However, when feedback is introduced into the network, the phase couplings and their coupling strengths determine how quickly oscillators phase-lock. To illustrate this, the two lateral ankle oscillators are coupled through 4 other oscillators, and a shift in phase in the left lateral ankle oscillator will take some time to synchronize to the right lateral ankle oscillator.

4.3 Driving the Robot

The parameters μ_i, ω_i and phase difference ψ_{ij} can be directly taken from the evolved sine controller parameterization. To get the exact same behavior as the sine controller, the initial values of the state variables of the oscillator should be set so that the oscillator is exactly on the limit cycle. In the phase space representation, the initial values of r_i and ϕ_i are simply set to μ_i and ψ_{ij}. In Cartesian space, the following initial values for x_i and y_i put the oscillator on the limit cycle:

$$\begin{aligned} x_i &= \mu_i \sin(\phi_i) + C_i \\ y_i &= \mu_i \sin(\phi_i + \frac{\pi}{2}) \end{aligned} \tag{4.12}$$

Because each oscillator is already on the limit cycle, the coupling between the oscillators is effectively of no influence. Therefore, in this case, the coupling weight β_{ij} can be set to any arbitrary value when trying to get the exact same behavior as the sine controller. In this case it is set to 1. The parameters that determine the speed of convergence (γ_x, γ_y or γ_r) are set to 15. This value was experimentally determined to have the oscillators match the sine controller well with an integration step of the dynamical system of 1ms. All oscillators are integrated using Euler integration.

4.4 Using the CPG's Intrinsic Properties

The start of the walk sequence, in open-loop, is very influential for a certain gait to be stable. Although a gait might be stable in open-loop in itself, given a specific set of initial conditions the robot might never reach the stable region of locomotion if the oscillators are simply turned on as with the sine controller. The robot is likely to not be able to walk at all under different starting conditions, even more so since the initial upright starting posture is not actually part of the normal gait cycle. This means that when you would initialize the oscillators on the initial robot joint positions, they would not be on the limit cycle.

To illustrate the advantages of the CPG as a controller methodology, an experiment is deviced that uses the intrinsic properties of the oscillator network to provide the necessary smooth transition from the initial posture of the robot to the stable gait. Given that it is not clear how parameters should be chosen to acquire the correct behavior, an evolutionary

algorithm is used to search for solutions in the parameter space of the CPG.

4.4.1 Optimizing Start of Locomotion

Due to its simple definition, and since there is no need for feedback integration for this experiment, the oscillator definition in phase space is used. The first issue is how to put the current state of the oscillator for each joint at the initial posture, such that $x = x_0$. An intuitive way to solve this issue is to set $r = 0$, and use $c = x_0$. This means that the initial amplitude of each oscillator is set to 0 and is then scaled up (with a rate related to γ_r) to the final amplitude. Similarly, the offset of each oscillator will converge towards the encoded offset of the gait with a rate related to γ_c. The two gain parameters γ_r and γ_c are open and subject to the optimization process. Since the gait is symmetric, it is sensible to limit the search area to evolve the γ_r and γ_c gains for the following groups of joints (resulting in a total of 8 parameters):

- Lateral hip and ankle joints
- Frontal hip joints
- Frontal ankle joints
- Knee joints

The initial values of the oscillator phases are set so that the oscillators within each of the legs are initially phase-locked with respect to the phase of the leg (as defined by the front hip oscillator of each leg). The phase difference of the left and right leg is considered an open parameter with the constraint that the phase is set such that the robot will always start with the left leg being the first leg in stance phase. This means that the left leg always starts in a backward motion ($\phi = [-\frac{\pi}{2}, \frac{\pi}{2}]$).

The coupling weight of the coupling between the legs is also considered an open parameter to be optimized (to allow for different speeds of convergence to phase-lock the legs). The coupling between the oscillators within each leg (β_{inter}) is set fixed to 50 to ensure fast convergence to any change in phase caused by the synchronization of the two legs.

A summary of the open parameters can be found in table 4.1, with β_{inter} the coupling weight of all oscillator connections except for the connections between the left and right

frontal hip joints. The total number of open parameters is 10.

Table 4.1: Phase Space Optimization Parameters

Par.	**Boundaries**
$\psi_{lr} = -\psi_{rl}$	$[0,\ \pi]$
$\beta_{lr} = \beta_{rl}$	[0, 50]
γ_r	[0, 50]
γ_c	[0, 50]

For this experiment a set of six initial postures are chosen such that the postures vary significantly in terms of the position of the center of mass of the robot and initial joint angles. The postures were determined experimentally. The initial upright posture is included in the six postures. The experiment was run with the optimized sine controllers, translated to the CPG network, for all four frequency gaits. The robot was brought to its initial position in the first 2 seconds of the experiment, whereafter the CPG network was activated. The same PSO algorithm with the same settings as was used for the optimization of the nominal gait is used. Each run lasted 30 seconds and the fitness function is the mean walked distance over the six runs.

Results

The progress of the PSO algorithm is similar to that of the nominal gait optimization as can be seen in figure 4.2. The process shows good convergence to the (sub)optimal solution.

The optimized values for the open parameters are displayed in table 4.2. The resulting wave patterns for the 0.75 Hz gait are shown in figures 4.3 to 4.6 (here the lateral ankle patterns are omitted since they are in exact anti-phase of the lateral hip joints). The wave patterns for the other frequencies can be found in the appendix (figures A-1 to A-8) for reference. Videos showing the optimized behaviors can be found at:

`http://birg2.epfl.ch/~jvanden/videos/thesis/start_walk/`.

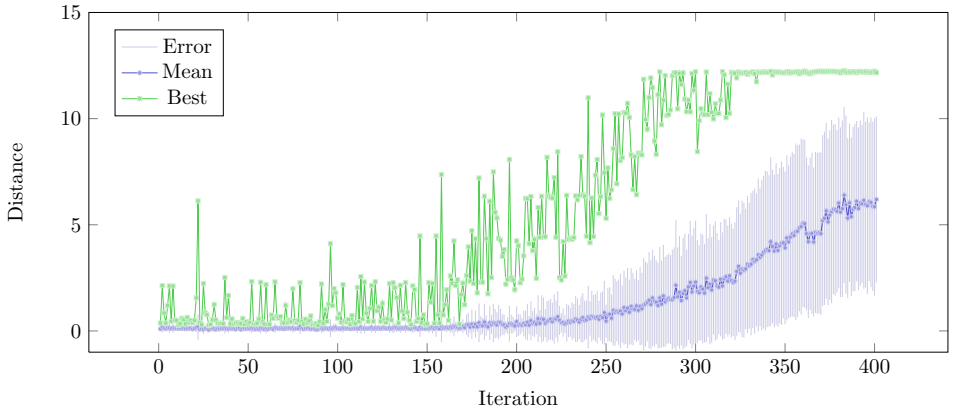

Figure 4.2: Progress of optimization process

Table 4.2: Evolved parameter values for startup experiment

			Lat. Hip		Front Hip		Knee		Front Ankle	
Freq.	ψ_{lr}	β_{lr}	γ_r	γ_c	γ_r	γ_c	γ_r	γ_c	γ_r	γ_c
0.50 Hz	3.16	41.03	49.96	24.89	10.24	47.71	15.11	48.20	0.00	45.02
0.75 Hz	3.14	0.31	49.97	34.46	14.41	35.11	45.23	28.56	45.97	47.25
1.00 Hz	3.09	16.96	50.00	50.00	3.71	34.62	4.66	29.32	8.12	37.77
1.25 Hz	3.14	34.48	41.19	34.82	17.24	45.42	25.46	50.00	36.45	50.00

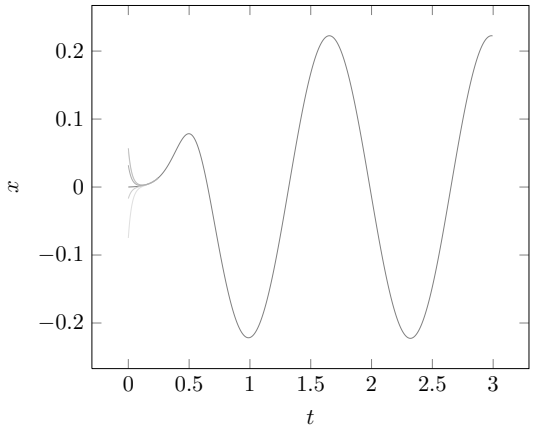

Figure 4.3: Lateral Hip

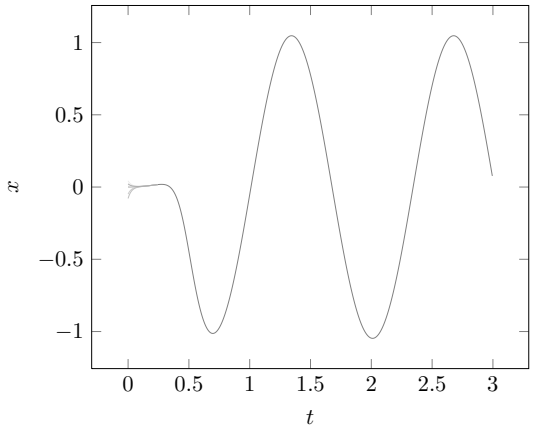

Figure 4.4: Frontal Hip

Discussion

The resulting wave patterns for the four different frequency gaits show similar characteristics. The knee and the front ankle joints show fast convergence in the offset. This

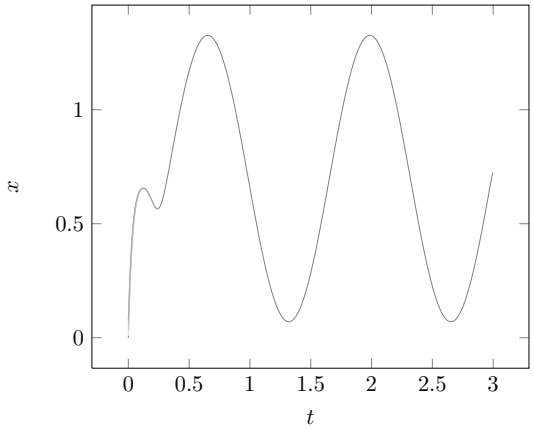

Figure 4.5: Knee

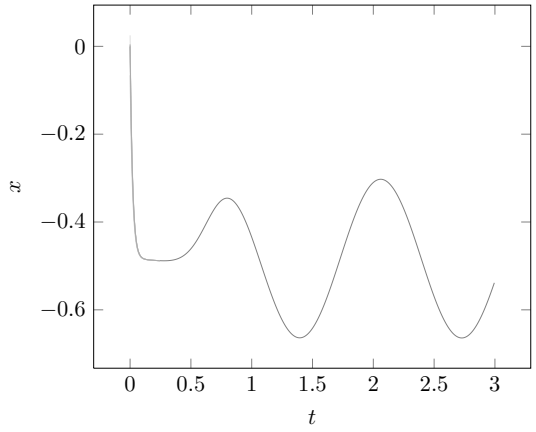

Figure 4.6: Frontal Ankle

means that the robot will first move towards a more crouched, stable position. Hereafter, the lateral joints will start to move towards one side to shift the load of the body to one leg. Finally, when there is enough load on one leg, the other oscillators will start (with increasing amplitude). The 1.00 Hz gait shows a longer period to start the oscillation due to a small gain of the amplitude. This results in a more gentle start of the gait than the other frequencies.

These findings are reflected in the optimized values directly. It can be seen that indeed, the gains for convergence to the desired amplitude is fairly small for the frontal hip joints, causing the robot to "wait" before actual forward motion is started. It can also be seen that it is favoured for both legs to be phase-locked (difference of π) from the beginning.

The resulting network is able to go towards a stable gait for the six different initial conditions, as well as generalize beyond these conditions. It should be noted that the best solutions can show small unstable situations in start of the walk while less optimized solutions might show a more stable start, but are not able to generalize as well. These issues can be solved with the use of sensor information to provide better balance of the robot during this task.

The results as outlined show that even in open-loop the intrinsic properties of the CPG can provide very useful characteristics. Here it is shown that without feedback and with proper settings of the CPG parameters, a smooth start of the locomotion can be achieved. The result of the optimization process matches very well the most important requirement of starting to walk, namely shifting the weight of the body to one leg to safely start the swing phase with the other leg.

Chapter 5

Stability

In the previous chapters a general method for designing a nominal gait for a humanoid robot was described. Furthermore, it was shown that CPGs provide a very interesting, flexible and dynamic model for designing a controller for which intrinsic properties can be used to provide stabilizing properties to the gait without feedback. This last chapter will discuss the types of sensor information used by humans and current biped robots. After this brief discussion, a framework will be defined and validated to measure stability or *instability* in a quantative way.

5.1 Stability

Since the physical structure of the robot is modeled after the human body, a good starting point for analyzing how information from sensors is used, combined and integrated, is to look at the biological mechanisms in humans that accomplish this. Although on a low level, the physical properties of the human body are very different from the properties of the robot, some mechanisms could still apply to both.

One interesting mechanism is that of the alpha/gamma motor system. Here information on muscle stretch/length originating from muscle spindles is influenced by the activity of gamma motor neurons. This activity includes information from cutaneous, muscle and articular tissues. This shows that the information resulting from the muscle spindles is not only dependent on the actual muscle length, but also actively modulated (Riemann & Lephart, 2002b, 2002a).

One of the extensively researched mechanisms for stabilization are reflexes (Duysens

et al., 1990; Yang & Stein, 1990; Zehr & Stein, 1999). In short, reflexes can be grossly divided into cutaneous and muscle (stretch and load) reflexes based on their function for stabilization during locomotion. Furthermore, different reflexes are found to be active during different parts of the gait cycle. Four different conditions can be defined:

1. *Swing phase*

   In the swing phase, the contact of the foot with an object produces a stumble correction by facilitating complex reflexes in both the stance and swing leg. The reaction decreases the stiffness of the swing leg so that the swing leg can continue its normal trajectory to foot contact. For non cutaneous perturbations, muscle stretch reflexes acting on the knee are produced to maintain the swing motion.

2. *Swing to stance phase*

   Both cutaneous and muscle reflexes are important in placing reactions to ready the stance limb for weight support during the transition. Sensation of the ground plays an important role in placing the foot.

3. *Stance phase*

   Load receptor reflexes act to control the step cycle timing and contribute to force production needed for forward propulsion during the stance phase. Cutaneous reflexes help to stabilize the stance phase on an uneven surface by altering the ankle trajectories. Muscle reflexes act in weight support and stabilization and load receptor reflexes affect cycle timing and support.

4. *Stance to swing phase*

   Muscle and load receptor reflexes sense unloading of the leg and affect cycle timing. If there is a slip, stance leg ankle muscles are stretched and ground contact will be prolonged. Stance termination is signaled by muscle and load receptor reflexes. Cutaneous reflexes will act to withdraw the leg from the ground.

5.1.1 Bipedal Robots

In robotics there are a few mechanisms that are very widely used in stabilizing biped locomotion. The first and probably most well known method is manipulation of the ZMP

(Zero Moment Point) (Vukobratovic & Juricic, 1969). The ZMP is a point on the ground where the resultant of the ground reaction force acts. Stable locomotion can be achieved when the ZMP is within the convex hull of the support base of the robot. The ZMP criteria is often used in path planning to ensure a certain motor pattern is ensured to result in a stable gait. This method requires the full model of the robot and the environment to be known.

The *concept* of the ZMP can also be used without having explicit knowledge of the model. By simply using load sensors at the sole of the foot, instability criteria can be produced, similar to the ZMP criteria. Although one of the major problems with the ZMP is that it does not work on surfaces that are not flat, work has been done to overcome this problem by introducing virtual surfaces (Sugihara et al., 2002). For a humanoid robot, direct manipulation of the ZMP point is not possible. Therefore, control laws have to be found which indirectly influence the ZMP so that it stays within the stable region. Both the upper body as well as the hip and ankle joints can be used to accomplish this.

A different kind of criterion for instability was introduced in (Goswami, 1999). Here a different point, called the Foot Rotation Point (FRI), is used from which the rotational force on the foot can be determined. Consequentially, appropriate action can be taken when rotation of the foot is detected. The advantage of this approach is that unlike the ZMP, the FRI can be outside the foot support area and when it does so, the distance to the support area indicates the severity of instability.

CPG based approaches often also adopt a strategy which is inspired on reflexes found in the human locomotion system (as described earlier) to stabilize the gait using the phase of the gait (Kun & Miller III, 1999; Aoi & Tsuchiya, 2005; Righetti & Ijspeert, 2006; Nakanishi et al., 2004). Here both cutaneous and load sensors can be used to detect whether a phase transition should occur and modulate the oscillator appropriately. For instance, when there is still load on the stance leg, it should not go into swing phase. Similarly, when ground contact is detected on the leg which is in swing phase, a fast transition to stance phase should occur.

In force/torque controlled robot systems, it has been shown that a model of the stretch reflex can be used to entrain locomotion to the mechanical properties of the robot (Taga, 1998; Taga et al., 1991). Also, simple feedback mechanisms using load sensors on the sole of the foot can be used to entrain natural locomotion frequency resulting from the robot

mechanics (Morimoto et al., 2006; Nakanishi et al., 2004).

In (Park & Kwon, 2001), reflex like mechanisms are used to stabilize a robot when it encounters a slippery surface by ad-hoc strategies including lifting of the hip to increase the force of the robot acting on the ground (effectively increasing the friction of the foot with the ground) and moving the foot of the swing leg towards the stance leg to prepare for taking over the load of the body.

5.1.2 Summary

A stable solution for biped locomotion should be able to walk in an unpredictable environment including slopes, small objects, surfaces with varying friction properties, unstable starting conditions and perturbations by external forces such as wind or somebody pushing the robot. To accomplish this for a rigid robot with no compliant elements, using a CPG based controller, inspiration should be taken from biological systems where possible. Reflexes play an important role in providing mechanisms for stabilization resulting in phase correction, ZMP manipulation, recovering from hitting small obstacles, etc. Entrainment to mechanical properties and learning capabilities in general are important to be able to generalize stabilization properties over a large range of parameter settings and even different robots.

5.2 Measuring Stability

Being able to measure the quality of a gait in terms of stability in an objective, quantitative and standardized way is important. Such a method could be used as a fitness criteria for optimization algorithms, feedback control and is also important to make quantitative comparisons with other research possible. This section describes a newly developed, practical framework in which stability of a biped gait can be quantitatively measured in a simulation environment such as Webots. The method is independent of the type of robot or controller and as such is not limited to the HOAP-2 humanoid robot as used throughout this thesis.

The following sections will shortly introduce common concepts in multi-body systems, and in particular bipedal systems, such as the Foot Support Area, Center of Mass, Ground Projection of the Center Of Mass and Center of Pressure. Thereafter, several well known

and widely used criteria/methods for stabilization will be discussed: the Zero Moment Point, Foot Rotation Indicator, Centroidal Moment Pivot and Poincaré maps. Finally a general method for measuring stability will be described that combines beforementioned criteria for stability in a general way. In the remainder of this section, the ground surface is assumed to be planar.

5.2.1 Foot Support Area

The foot support area of a bipedal robot, on planar ground, is defined as the convex hull of the contact points of the robots feet. Stability indicators are commonly defined as points on the ground which ensure the robot stability as long as they are inside the foot support area.

5.2.2 Center of Mass

The center of mass OG (defined in coordinate system O) of a robot, composed of n segments denoted by i, is defined as:

$$M = \sum_{i=0}^{n} m_i \tag{5.1}$$
$$OG = \frac{1}{M} \sum_{i=0}^{n} OG_i \cdot m_i$$

where OG_i and m_i are the center of mass (defined in coordinate system O) and the mass of the ith segment respectively. Furthermore, the ground projection of the center of mass, or GCoM, is defined as the projection along the gravity axis of the center of mass onto the ground. The position of the GCoM can be used as a static stability criterion (e.g. posture control) where the robot is said to be stable as long as the GCoM is within the convex hull of the foot support area. It can be easily seen however that this does not extend to a dynamic criteria. Indeed, when walking, the GCoM will have to leave the support area for the robot to move forward.

5.2.3 Center of Pressure

The Center of Pressure is defined as the point on the ground where the force, resultant from the field of pressure forces normal to the foot, is exerted and the resultant moment

is zero. The Center of Pressure and the Ground Projection of the Center of Mass coincide when the robot is stationary or has uniform linear and angular velocities in all joints. It can be defined as:

$$OP = \frac{\sum_i q_i \cdot F_{ni}}{\sum_i F_{ni}} \tag{5.2}$$

Where OP is the Center of Pressure defined in coordinate system O, q_i is the vector to the point where force F_{ni} acts in the direction of the surface normal.

5.2.4 Zero Moment Point

The Zero Moment Point (Vukobratovic & Juricic, 1969), or ZMP, is a widely used criteria (both implicitly and explicitly) for dynamic stability in biped locomotion. The Zero Moment Point is defined as the point on the ground at which the moment generated by the reaction forces and torques equals zero. It is stated that this point must lie within the convex hull of the foot support area to ensure stability. The ZMP in its general form is most widely used for trajectory generation and is defined as:

$$PZ = \frac{n \times M_P^{gi}}{F^{gi} \cdot n} \tag{5.3}$$

Here, P is point on the sole of the foot following the normal projection of the ankle, Z is the ZMP, n is the normal vector from the surface, M_P^{gi} is the moment about point P and F^{gi} is the resultant of gravity plus inertia forces.

Even though the most common use of the ZMP is in trajectory planning, the general idea of the ZMP is often also used in on-line, reactive stability control. As shown in (Goswami, 1999; Sardain & Bessonnet, 2004), the Center of Pressure is equal to the ZMP. Due to the definition of the ZMP, it can never leave the foot support area. This means that it cannot distinguish between a marginally stable equilibrium and a unstable situation since both states are located at the boundary of the support polygon. In other words, it can be used as a criteria for stability, but not for *instability*. The Foot Rotation Indicator tries to overcome this problem.

5.2.5 Foot Rotation Indicator

The Foot Rotation Indicator, or FRI, is an indication of postural instability measured as the the point on the foot/ground where the net ground forces would have to act to keep the

foot stationary (Goswami, 1999). Stability with regard to foot rotation is ensured when the FRI point remains within the convex hull of the foot support area. The difference with the ZMP as described above is that the Center of Pressure represents the point on the foot/ground where the net ground reaction force actually acts and may never leave the convex hull of the foot support. The FRI point on the other hand is defined as the resultant moment of force/torque impressed on the foot is normal to the surface and *can* leave the convex hull of the foot support. The distance from the area of support indicates of the severity of unbalanced torque causing the foot to rotate.

The Foot Rotation Indicator can be calculated by using equation 5.4 (Goswami, 1999). Here the FRI (F) is calculated in coordinate system O. In this equation N is the total number of segments in the multi-body robot (*excluding the foot*), m_i is the mass of segment i, a_i is the linear acceleration of segment i and $\dot{H}_{OG_i}$ is the angular momentum about the center of mass OG_i, defined in coordinate system O, of segment i.

$$
\begin{aligned}
OF_x &= \frac{m_1 OG_{1y} g + \sum_{i=2}^{N} m_i OG_{iy}(a_{iz}+g) - \sum_{i=2}^{N} m_i OG_{iz} a_{iy} + \sum_{i=2}^{N} \dot{H}_{G_{ix}}}{m_1 g + \sum_{i=2}^{N} m_i (a_{iz}+g)} \\
OF_y &= \frac{m_1 OG_{1x} g + \sum_{i=2}^{N} m_i OG_{ix}(a_{iz}+g) - \sum_{i=2}^{N} m_i OG_{iz} a_{ix} - \sum_{i=2}^{N} \dot{H}_{G_{iy}}}{m_1 g + \sum_{i=2}^{N} m_i (a_{iz}+g)}
\end{aligned}
\tag{5.4}
$$

5.2.6 Centroidal Moment Pivot

One of the problems with the FRI is that it is only defined during the single support phase since it is defined in terms of rotation of the foot. The Centroidal Moment Pivot overcomes this problem (Popovic, Goswami, & Herr, 2005). The CMP is based on the finding that the body's angular momentum about its center of mass remains very small throughout the gait cycle. As such, the angular momentum can be used to determine robot instability.

Given that the ground is planar, the CMP is defined as the point where a line parallel to the ground reaction force, passing through the center of mass of the robot, intersects with the ground. As such, it provides information on body rotation. The CMP can be

written in terms of the center of mass and the ground reaction force as:

$$C_x = \mathrm{CoM}_x - \frac{F_{\mathrm{GR}_x}}{F_{\mathrm{GR}_z}}\mathrm{CoM}_z \tag{5.5}$$
$$C_y = \mathrm{CoM}_y - \frac{F_{\mathrm{GR}_y}}{F_{\mathrm{GR}_z}}\mathrm{CoM}_z$$

where C is the CMP position, CoM is the center of mass and F_{GR} is the ground reaction force.

5.2.7 Poincaré maps

Poincaré maps are often used in analyzing continuous, periodic dynamical systems. A Poincaré map of an N dimensional state space defines a mapping from the N-1 dimensional state space at a Poincaré section onto itself. Here a Poincaré section is a hyperplane in the state space which is traverse to the flow of the dynamical system. In stability analysis the periodic motion is defined as stable when the eigenvalues of the linearized Poincaré map (the Jacobian matrix) are within the unit cycle (e.g. the system will converge to a fixed point on the Poincaré section).

In biped locomotion, this method is often used to quantify the stability of a dynamical system controlling the different joints. However, to be able to derive the stability properties, one has to know or approximate the Poincaré map and the Jacobian matrix of the full dynamical system. It is therefore often used when the exact model of the robot and the environment is known (usually expressed as equations of motion) (Morimoto et al., 2005), or to derive the stability of only the controller, but not in interaction with the environment.

5.2.8 Measuring General Stability

Although the Poincaré map method is often not applicable in practical situations where there are unknowns about the environment and the model of the robot, the concept can still be applied to measure stability. For any given bipedal periodic gait, a Poincaré section can be defined as a certain point in the gait cycle. Given any state variable of the biped robot, it being a state variable of the controller, or a measured quantity such as center of pressure, the assumption can be made that stability is directly related to the variance

of the distribution of state variables intersecting the Poincaré section during locomotion. The more unstable the gait is, the larger the variance in certain state variables will be.

This results in a very simple method to evaluate stability of any measurable quantity that can be evaluated in simulation while walking given any controller. Examples of measurable quantities are the amplitude and phase of a coupled oscillator network, the load distribution of the robot as measured with load sensors in the feet, body posture measured by a gyroscope or body acceleration measured by an accelerometer. Furthermore, the approximate ZMP, FRI and CMP can also be used in the same way. It should be noted that this method gives an objective measurement of stability independent of the type of controller, and as such can be applied to compare for example different feedback control loops in terms of stability of the gait.

Other notions of stability can also be taken into account here. It might be preferable that the upper body of the robot remains in an upright posture. To this extend, a criterion can be defined that uses the posture of the upper body directly and defines instability as the deviation of the upright position of the body. A similar measurement of stability can be taken from body acceleration. In a ideal forward bipedal gait, the acceleration of the upper-body should be 0 (e.g. walking with a constant speed). Deviation from this can be directly used as an instability measurement.

It should be noted that depending on the information available to the robot, these measurements of stability or instability, might only by used to evaluate a certain gait/controller, but not for active stabilization. An example of this is the FRI which the robot might not have direct access to or be able to calculate. The CoP on the other hand can often be easily calculated when the robot has force sensors at its feet. Thus, there is a clear distinction between measurements that can be used as feedback into the controller, and measurements that can be only used for evaluating a certain gait or controller.

5.2.9 Practical Framework

The sections above have described different existing stability criterion and have put forth a general method of combining any of these criteria in a general measure. This section describes how to use the Webots simulator to implement all the mentioned criteria for any given robot modeled in the Webots environment. The measurements described below are implemented in Webots as a *physics plugin* instead of in the robot controller. The

reason for this is that many of the calculations require information that is not available from the controller (such as iterating over all the bodies of the robot and acquiring force measurements). A physics plugin has direct access to the physics simulation as done by ODE (the Open Dynamics Engine) which is used internally by Webots. Note that we use the convention of defining the z axis as the *up*-axis whereas in ODE terms, the y axis is the *up*-axis. Where appropriate, the HOAP-2 robot will be used as an example.

Center of Mass

Calculating the center of mass of the robot is straightforward from equation 5.1. The position of the center of mass of each body, and the mass of each body can be obtained from ODE. The coordinate system O is defined as the global world coordinate system.

The Ground Projection of the Center of Mass can furthermore be obtained by projecting the previously calculated Center of Mass along the gravity axis. Since we assume a planar ground, the projection on the ground is the intersection of the Center of Mass at $z = 0$.

Center of Pressure

To be able to calculate the Center of Pressure, or ZMP, we need access to the forces as applied to the contact points of the feet of the robot. We therefore need to manually apply the collision detection and store contact points resulting from this collision detection for later use. ODE allows force/torque measurements to be taken from these contact points in a later stage of the simulation.

The real HOAP-2 robot has four force sensors at each foot as depicted in figure 5.1. The actual sensors s_i are located away from the edges on the real robot. However, the contact model of ODE will generate contact points on the edges of the feet shown c_i for a planar ground surface. To roughly approximate the measurements that would be made on the real robot, the following equation is used:

$$F_{s_i} = \sum_{j=0}^{C_N} \left(\frac{1}{\delta(s_{i_{xy}}, c_{j_{xy}}) \cdot T_j} \cdot F_{c_j} \right) \tag{5.6}$$

$$T_j = \sum_{i=0}^{S_N} \frac{1}{\delta(s_{i_{xy}}, c_{j_{xy}})}$$

Here C_N is the number of contact points, S_N is the number of sensors (e.g. four in this model), δ is the euclidean distance and F_{c_j} is the force measured at contact point c_j.

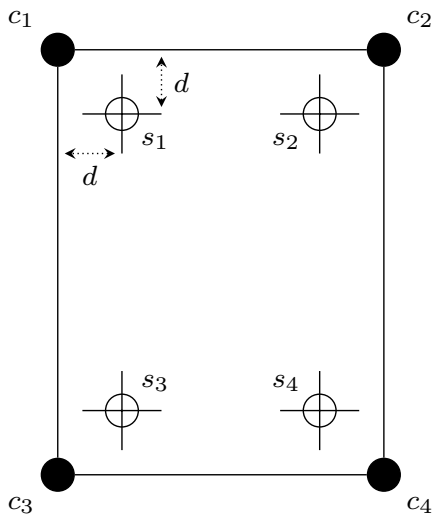

Figure 5.1: Schematic view of the feet of the HOAP-2 robot

The Center of Pressure can now be calculated from the forces as approximately measured by the sensors on the sole of the foot in the direction of the surface normal (e.g. [0, 0, 1]) as defined previously in equation 5.2.

Foot Rotation Indicator

The Foot Rotation Indicator as defined in equation 5.4 can be directly calculated from information present in the physics model. For each of the segments of the multi-body robot, the center of mass and the mass itself can be directly accessed. The linear acceleration of the CoM (a_i) can be either estimated by reversing the Euler Integration done by ODE on the current and previous velocities of the body, or by measuring all forces acting on the body and dividing it by the body mass. Finally, the angular momentum $\dot{H}_{OG_i}$ about the center of mass is defined as:

$$\dot{H}_{OG_i} = I_{OG_i} \cdot \omega_i \tag{5.7}$$

Here I_{OG_i} is the Inertia tensor matrix as defined in ODE for body i and ω_i is the angular velocity of body i which can be accessed from ODE.

Centroidal Moment Pivot

The Centroidal Moment Pivot can be estimated easily from previously determined quantities. Recall that from equation 5.5 that the CMP can be determined directly from the robots center of mass and the ground reaction force. The ground reaction force can be calculated by the sum of the forces generated at the contact points of the feet.

Readily Available Measurements

Some measurements can be retrieved from the Webots environment directly without resorting to accessing the physics simulation directly. These measurements include body posture (as exposed by the Gyroscope device) and body acceleration (as exposed by the Accelerometer).

5.2.10 Summary

In summary, the following measurements now can be used as state variables in the stability measuring framework:

- Center of Mass
- Center of Pressure (ZMP)
- Foot Rotation Indicator (FRI)
- Centroidal Moment Pivot (CMP)
- Body Posture (Gyroscope)
- Body Acceleration (Accelerometer)

Not all of these measurements can also directly be used to use for stabilizing the robot by feedback control. The FRI for instance requires information that might not be available on the real robot. Information that *is* readily available on most robots is CoP (ZMP), Body Posture and Body Acceleration. Any control laws resulting from these measurements can be quantitatively compared in simulation using the described framework.

5.3 Stability under Perturbation

To validate the described framework with regard to detecting instability due to external perturbations and general stability of a gait, an experimental setting is designed. In this setting an external force is applied at the middle of the torso of the robot while it is walking. The force is applied for a uniformly distributed random duration between 0.25 and 0.75 seconds, with uniformly distributed random pauses between 0 and 2 seconds. This results on average in the robot being perturbed one third of the time. The direction of the force is perpendicular to the torso in the *up* direction, with a uniformly randomly distributed angle (see figure 5.2).

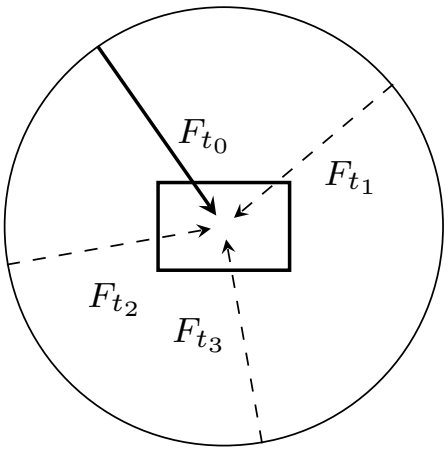

Figure 5.2: Schematic view of applying random external forces F at times t_0, t_1, t_2 and t_3.

By increasing the applied external force and measuring at each force level the measurements in the framework as previously described, we can measure how well the framework of measuring instability follows the increasing force. Here we expect the variance of good instability indicators to increase with increasing forces. The experiment is designed to:

1. Validate the instability measuring framework
2. Evaluate how well the different indicators measure instability
3. Evaluate how stable the evolved open-loop controllers are with regard to external perturbation

5.4 Experiment

For this experiment the evolved sine controllers are converted to a CPG network (see figure 4.1) using the Cartesian space Hopf oscillator as described in equations 4.10 and 4.11. The oscillators are all put on the limit cycle using equation 4.12. Since all oscillators are already on the limit cycle, and thus phase-locked, the phase coupling weights can be set arbitrarily (in this case they are set to 1). Choosing $\gamma_r = 15$ for each oscillator, it is ensured that the exact same sine wave is produced by the network.

Each run consists of 5 seconds startup time, enabling the gait to stabilize from starting the locomotion. Hereafter the perturbation at a certain force level is started and the robot is allowed to walk a maximum of 20 steps (where a step is defined as one half of a full gait cycle). Measurements are taken at 3 equally spaced intervals during each step (for each leg). If the full gait cycle is defined from 0 to 1, measurements are taken at 0.125, 0.25, 0.375, 0.625, 0.75 and 0.875. When the robot falls down or has made the 20 steps, the run is ended. For each force level, 100 runs are performed. This results in 1000 measurements for each interval and each measurement if all runs for a certain force level are successful. The applied external forces range from 0 to 5 Newton with a step size of 0.1 Newton.

The indicators used in this experiment are: four load sensors measuring ground reaction force at each feet, torso acceleration on x, y, z axis, angular velocity on x, y, z axis, center of mass in x, y, z and center of pressure, foot rotation indicator and centroidal moment pivot in x and y.

All of the position indicators (such as the COM, COP, FRI etc.) need a reference point from which to measure the distance. In this experiment, the method for taking a reference point at a certain time t_i is depicted in figure 5.3. At each point in time when a leg is half-way in stance phase, the center of the position of the foot is recorded. For both legs, a path can then be constructed by interpolating at each time interval these recorded positions. The reference point, Ref_{t_i} at time $t = t_i$ is then defined as the center of the interpolated positions L_{t_i} and R_{t_i}. Each of the indicators can then be decomposed in its individual components (x, y and when applicable z) relative to the coordinate system as defined by L_{t_i}, R_{t_i}.

The experiment is run for the 0.50, 0.75 and 1.00 Hz gaits.

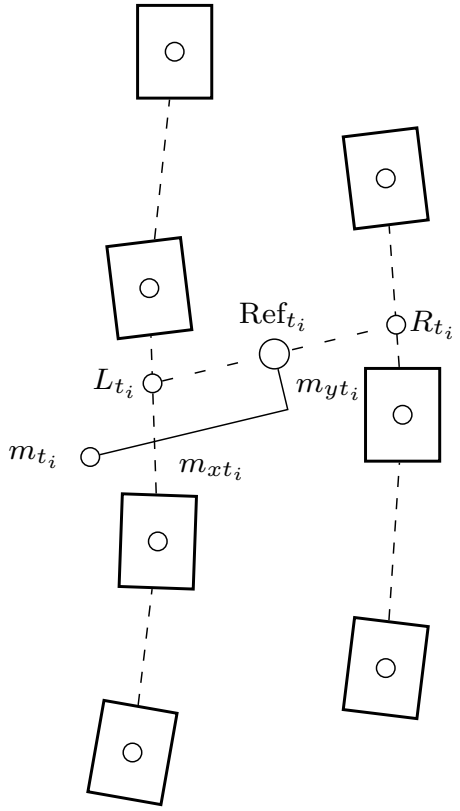

Figure 5.3: Schematic view of measuring distance at $t = t_i$ from reference point Ref_{t_i} to measurement m_{t_i}.

5.4.1 Results

Table 5.1 displays at what external force perturbation each frequency gait is able to walk a certain percentage (0.99, 0.9, ..., 0.25) of the time. This means that at for instance the 0.9 level, the robot on average is able to walk 18 steps before falling over. The forces are measured in Newton.

Table 5.1: Performance per Gait

Frequency	**0.99**	**0.95**	**0.9**	**0.8**	**0.5**	**0.25**
0.50	0.3	0.5	0.5	0.6	0.9	1.4
0.75	1.2	1.4	1.6	1.9	2.6	3.3
1.00	0.8	1.3	1.3	1.7	2.4	3.5

Figures 5.4 and 5.5 show the results of the stability measuring framework for the right

foot sensors of the 0.50 and the 0.75 Hz gaits. The distance to the foot support area for the CoP, FRI and CMP are shown in figures 5.6 and 5.7.

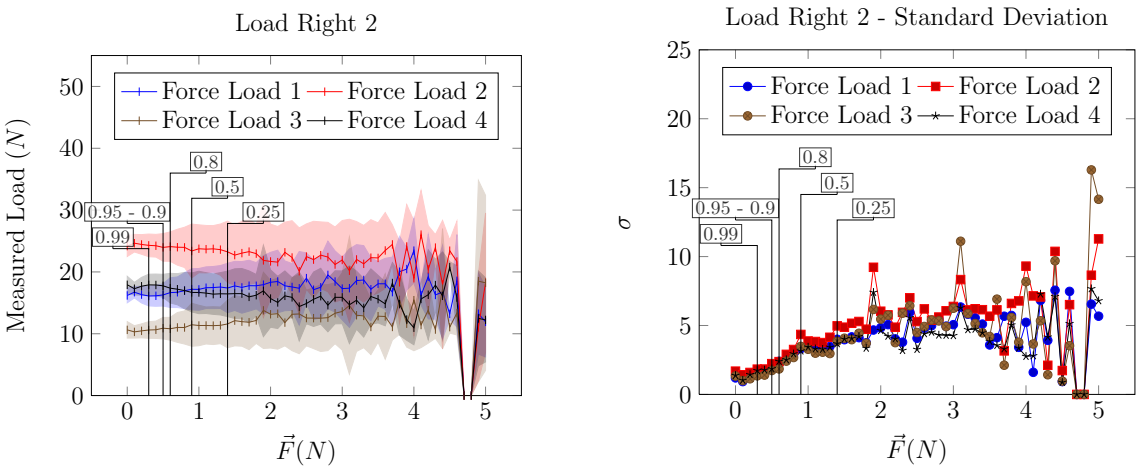

Figure 5.4: Force sensors of the right foot at 0.25 of the gait cycle (0.50 Hz)

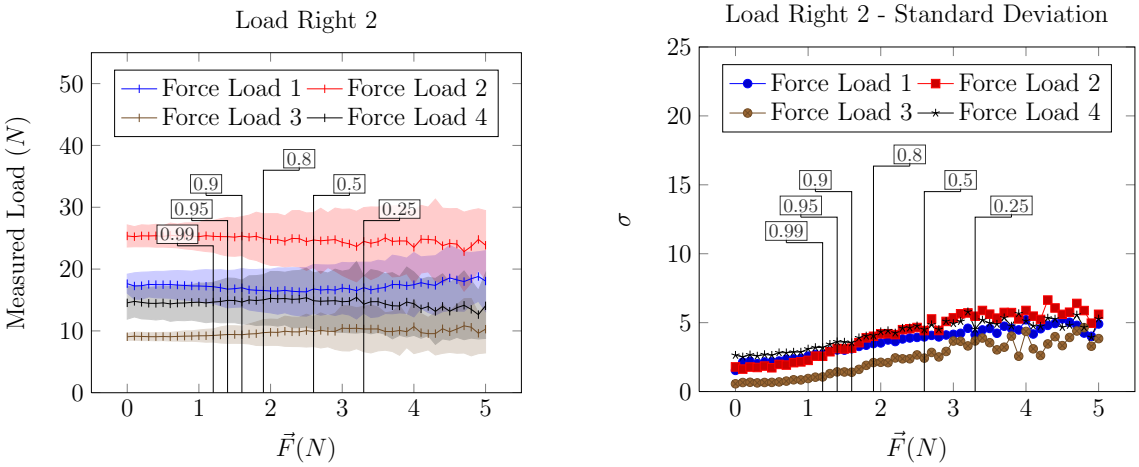

Figure 5.5: Force sensors of the right foot at 0.25 of the gait cycle (0.75 Hz)

Gait Quality

Table 5.1 shows that the 0.50 Hz gait performs the worst of all gaits, only able to walk for 0.95 of the time at maximum perturbation of 0.5 Newton. The 0.75 Hz gait proves to be the most stable, withstanding 1.4 Newton at the 0.95 level. Figures 5.4 and 5.5 display a clear difference in the behavior of the load sensors. For the 0.50 Hz the measured variance is larger and increases faster than for the 0.75 Hz gait. Similar results are found for the 1.00 Hz gait which also performs a little worse than the 0.75 Hz gait.

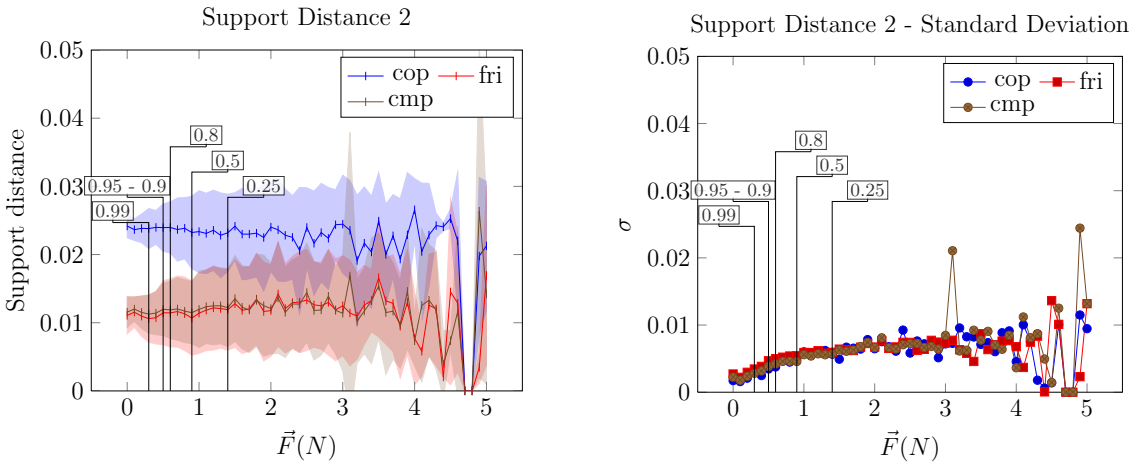

Figure 5.6: Support distance at 0.25 of the gait cycle (0.50 Hz)

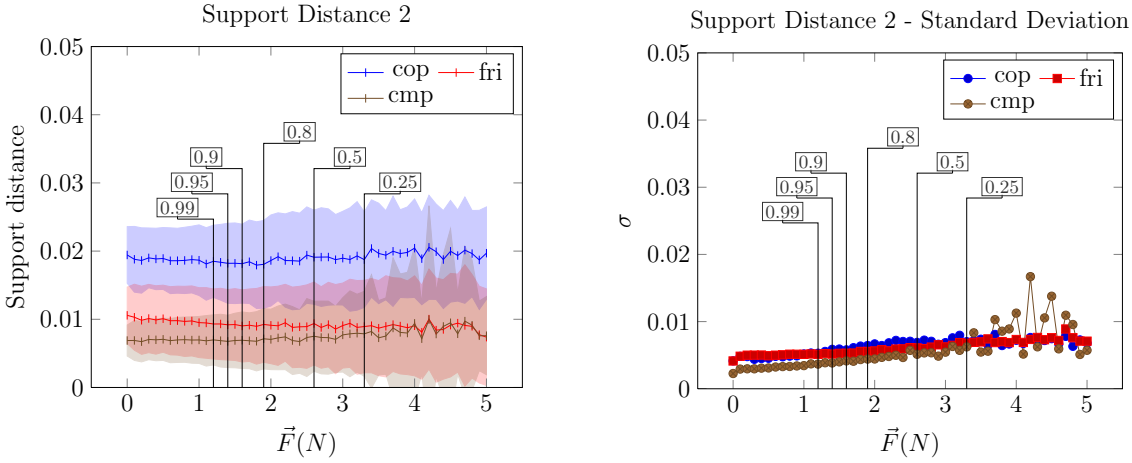

Figure 5.7: Support distance at 0.25 of the gait cycle (0.75 Hz)

Indicator Quality

The measurements at 0.125 and 0.385 of the gait cycle are noisier than the measurements at 0.250 of the gait cycle for the load sensors, angular velocity and acceleration. The center of mass, center of pressure, foot rotation indicator and centroidal moment pivot are less influenced by the stage of the gait cycle in which they are measured. There is a high correlation between the level of external force and the standard deviation for all of the measured quantities.

In general, the CoM, CoP, FRI and CMP stability indicators all perform quite equally. There are significant differences between the performance of each indicator type (with regard to how well it reflects the performance of the robot) between the different gaits.

For the 0.50 Hz gait, the foot load, acceleration and angular velocity measurements are much more descriptive than the CoM, CoP, FRI and CMP indicators. For the 0.75 Hz gait, this is less so the case. Here the acceleration and angular velocity reflect the change in performance much better. There is no single indicator that shows very clearly that at some force level the performance drops significantly.

5.4.2 Discussion

The experiment shows that the stability measuring framework performs as expected. Larger perturbations correlate with larger standard deviations in all measurements (e.g. the more perturbed the measurements become). On a global level, the experiment shows that stability can be expressed through this framework. It also shows that in a very practical sense, the framework can characterize the stability of a certain gait. What is not very clear is how this information could be utilized to prevent instability during locomotion. There is no single indicator which performs equally well for different types of gaits. Up to this point, each measurement is independent of the time at which it was taken during the 20 steps. This means that there is no information present on how a stability indicator may evolve during locomotion. Under perturbation, it can be expected that during locomotion the robot will become more unstable and this information with regard to the development of the stability indicator measurements could be used in feedback control.

Only one type of perburtation is currently tested, external perturbing forces. Other perturbations such as small obstacles, slopes or slippery surfaces could extend the framework to more ways of measuring the stability of a certain gait.

5.5 Active Stabilization

Now that there is a framework to measure gait stability in a quantitative way, the influence on the gait of introducing sensor information in the controller can be easily measured. To explore the increase in performance on the stabilization under perturbation, a simple feedback control loop is designed.

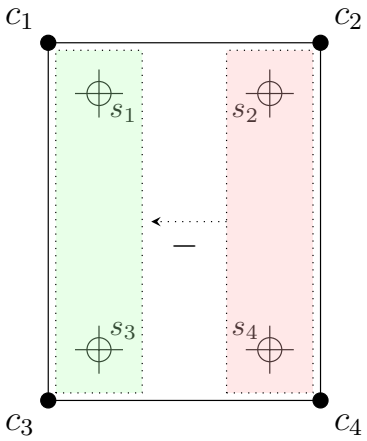

Figure 5.8: Lateral feedback

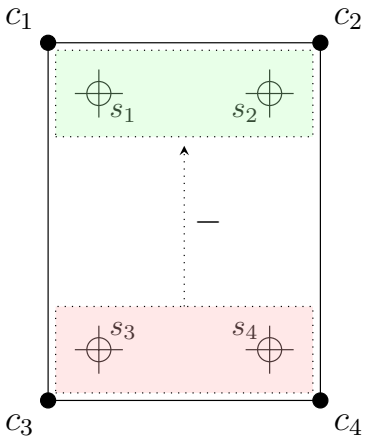

Figure 5.9: Frontal feedback

$$\epsilon_l = \frac{1}{W} \cdot ((s_1 + s_3) - (s_2 + s_4))$$
$$\epsilon_f = \frac{1}{W} \cdot ((s_1 + s_2) - (s_3 + s_4)) \tag{5.8}$$

Figures 5.8, 5.9 and equation 5.8 show how the foot sensors are used to provide information on respectively lateral and frontal load. ϵ_l and ϵ_f are respectively the lateral and frontal errors and W is the weight of the robot in Newton with which the measured sensor load is normalized. The result of this is that when there is more load on the left side of the foot, ϵ_l will be positive, and when there is more load on the right side of the foot, ϵ_l will be negative. The same holds for ϵ_f but with regard to load on the front and the back of the foot.

The errors ϵ_l and ϵ_f are used to adjust the ankle in order to move the load of the robot to the center of the foot. The feedback is applied to the ankle oscillator *offset*. This allows modification of the way load is distributed on the foot without altering the actual amplitude or phase of the oscillator. In order to facilitate simple learning capabilities of the oscillator, and allow entrainment with the environment, a small modification to the ankle oscillators is made. Recall the oscillator offset differential equation:

$$\dot{c} = \gamma_c(C - c)$$

If feedback would be applied on this equation, the offset would always try to go back to the original desired offset, whereas we would like to keep the newly *learned* offset

to stay encoded. In order to do this we make C part of the dynamical system and apply feedback on $\dot{C}$. Instead of using simple proportional feedback integration, we apply proportional-derivative feedback (as in a PD-controller). This introduces a damping factor in the system which prevents exponentially increasing oscillations. The equations for $\dot{C}$ therefore become:

$$\dot{C}_l = \gamma_l \cdot (\epsilon_l + \frac{\epsilon_l}{dt}) \tag{5.9}$$
$$\dot{C}_f = \gamma_f \cdot (\epsilon_f + \frac{\epsilon_f}{dt})$$

Here C_l and C_f are the lateral and frontal desired offset of the ankle oscillators respectively. γ_l and γ_f are two gain parameters which determine how quickly the desired offset is changed. The initial value of C is the desired offset as defined for the nominal gait. The feedback loop was tested in static conditions to see how well it would stabilize the load of the robot. The robot was put into a posture where the load was unevenly distributed and when the feedback was enabled, the load was quickly redistributed to the center of the support area.

The value of the gains is systematically explored in the same experimental setting as was used in the gait stability experiment. The experiment is run for the best open-loop gait as was measured before to be the 0.75 Hz gait. Values for the gains range from 0.001 to 0.01 with a step size of 0.0025 for the lateral offset, and 0.001 to 0.02 with a step size of 0.0025 for the frontal offset.

5.5.1 Results

Figures 5.10, 5.11 and 5.12 show the performance of the 0.75 Hz gait in the experimental setting as described above for respectively the 0.99, 0.95 and 0.90 success level. The performance is on average best with $\gamma_l = 0$ and $\gamma_f = 0.01$.

5.5.2 Discussion

The figures clearly show a significant increase in stability. The robot is able to reach performance levels under more perturbation. Although the lateral feedback can improve the stability of the gait, it seems the control of the frontal ankle joint is more influential in stabilizing this particular gait.

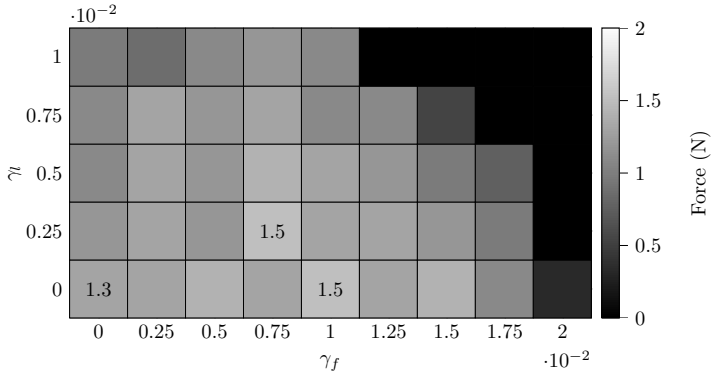

Figure 5.10: Maximum perturbation force for 0.99 success level

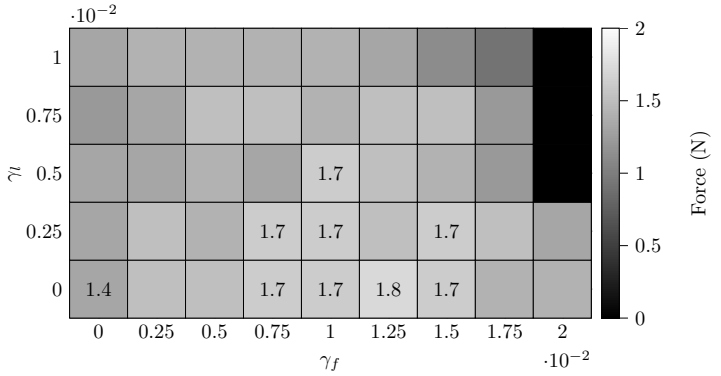

Figure 5.11: Maximum perturbation force for 0.95 success level

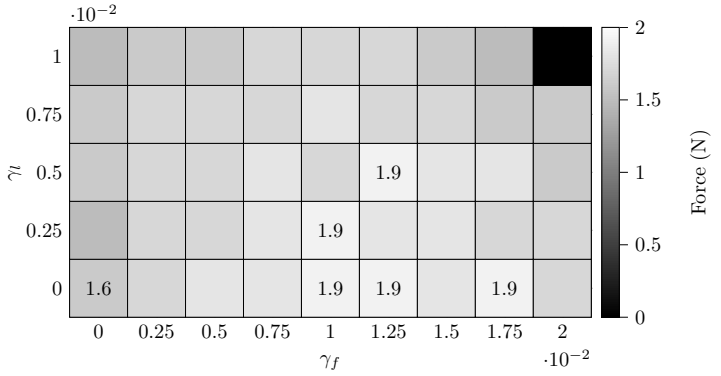

Figure 5.12: Maximum perturbation force for 0.90 success level

The results indicate that a simple feedback loop can be used to stabilize the gait, but it seems the type of feedback is too naive to result in large improvements. Although it confirms that feedback, though simple, can improve the gait stability, further work on different types of feedback integration such as reflexes and phase resetting as described in the beginning of this chapter is clearly needed.

Chapter 6

Conclusion

A PSO algorithm was successfully used to find stable open-loop nominal gaits for the humanoid HOAP-2 robot. Although a very simple sine controller was used to generate the joint angle patterns, the gaits looked very natural. Due to the fitness function being the walked distance, the gaits were optimized for maximum speed which made them very sensitive to parameter and environment changes. This also meant that low frequency gaits made very large steps, which also contributed to them only being on the limit of stable. In a sense it pushed the maximum speed an open-loop controller can achieve to the limit. This made it impossible to transfer the gaits to the real robot. The success of the method can be attributed to minimizing the parameter space by making informed decisions on the parameter constraints.

A network of oscillators was designed which encoded the evolved sine controller accurately. The Hopf oscillator defined in Cartesian space was extended to allow arbitrary and *independent* phase coupling. This allows easier integration of possible feedback than using the phase space Hopf oscillator without the disadvantage of influencing the amplitude of the oscillator. Furthermore it was shown that using the intrinsic properties and characteristics of the CPG, the stability of starting to walk could be improved without any feedback. This confirms that CPGs can be very useful for encoding behaviors implicitly without the need of specific rules or states. The produced behavior is also very natural and resembles very well the way load needs to be transfered to the stance leg before walking should begin.

Finally, a framework was developed in which stability of gaits could be quantified in a very practical and robot/controller independent manner. The stability measuring

framework proved to be successful in identifying which of the evolved gaits was the most stable. This provides a very generic and easy to use paradigm to quantify, compare and evaluate gait stability, and feedback control rules. The method is not limited to a specific robot or controller and as such can be used in a very general sense. Well known stability indicators such as ZMP, FRI and CMP were evaluated within this framework and no significant differences in these indicators with regard to gait stability under external perturbation were found.

6.1 Future work

There are many interesting questions and issues still to be explored. One in particular is the fitness function of the optimized nominal gait. Since the nominal gait is used as a basis for all other work, such as measuring and improving stability, it is a critical stage of designing a locomotion controller. In this thesis, a very simple and naive fitness function was used: walked distance. Although the PSO algorithm worked very well, and was able to find solutions that looked very natural, the solutions were highly optimized on speed. This means that they were very explosive and on the border of being stable. Any slight alteration or perturbation would make the gait unstable. This proved it to be impossible to transfer the gait to the real robot. A more sophisticated fitness function could address this issue and evolve a more robust and stable gait. This allows easier transfer to the real robot and ease the development of stability control mechanisms. Punishments and rewards incorporated in the fitness function could be desired stride length, speed restriction or stability measurements. Apart from changing the fitness function it could also be interesting to explore more complex joint angle patterns.

With regard to the CPG, more work can be done on the influence of different coupling schemes and oscillator parameter settings for coupling and convergence after perturbation. It has been shown that with proper parameter settings the CPG can encode a natural startup behavior. Further investigation on how to go from a nominal gait encoded in the CPG to more complex behaviors such as a start-walk-stop and making turns is needed.

Finally more work is needed on exploring integration of feedback into the CPG. A very simple feedback loop was shown to improve the stability of the gait, but there are many more possibilities to stabilize the gait further. Here inspiration should be

taken from mechanisms found in humans (such as reflexes and feedback dependent phase switching). The stability measuring framework could be extended to incorporate more types of perturbation such as obstacles, slopes and slippery surfaces.

Bibliography

Aoi, S., & Tsuchiya, K. (2005). Locomotion control of a biped robot using nonlinear oscillators. *Autonomous Robots*, *19*(3), 219-232.

Bastiaanse, C., Duysens, J., & Dietz, V. (2000). Modulation of cutaneous reflexes by load receptor input during human walking. *Experimental Brain Research*, *135*(2), 189–198.

Buchli, J., Iida, F., & Ijspeert, A. (2006). Finding resonance: Adaptive frequency oscillators for dynamic legged locomotion. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3903-3909.

Buchli, J., Righetti, L., & Ijspeert, A. J. (2005). A dynamical systems approach to learning: a frequency-adaptive hopper robot. *Proceedings of the VIIIth European Conference on Artificial Life ECAL 2005*, 210-220.

Buchli, J., Righetti, L., & Ijspeert, A. J. (2006). Engineering entrainment and adaptation in limit cycle systems. *Biological Cybernetics*, *95*(6), 645-664.

Collins, J., & Luca, C. (1993). Open-loop and closed-loop control of posture: A random-walk analysis of center-of-pressure trajectories. *Experimental Brain Research*, *95*(2), 308–318.

Collins, J. J., & Richmond, S. A. (1994). Hard-wired central pattern generators for quadrupedal locomotion. *Biological Cybernetics*, *71*(5), 375-385.

Collins, S., Ruina, A., Tedrake, R., & Wisse, M. (2005). *Efficient bipedal robots based on passive-dynamic walkers.* American Association for the Advancement of Science.

Dickinson, M. H., Farley, C. T., et al. (2000). How animals move: An integrative view. *Science*, *288*(5463), 100.

Dimitrijevic, M. R., Gerasimenko, Y., & Pinter, M. M. (1998). Evidence for a spinal central pattern generator in humans. *Annals of the New York Academy of Sciences*, *860*(1), 360.

Duysens, J., Trippel, M., Horstmann, G., & Dietz, V. (1990). Gating and reversal of reflexes in ankle muscles during human walking. *Experimental Brain Research*, *82*(2), 351–358.

Goswami, A. (1999). Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point. *The International Journal of Robotics Research*, *18*(6), 523.

Grillner, S., & Wallen, P. (1985). Central pattern generators for locomotion, with special reference to vertebrates. *Annual Review of Neuroscience*, *8*(1), 233-261.

Hooper, S. L. (2000). Central pattern generators. *Current Biology*, *10*(5), 176-179.

Ito, S., Yuasa, H., Luo, Z., Ito, M., & Yanagihara, D. (1998). A mathematical model of adaptive behavior in quadruped locomotion. *Biological Cybernetics*, *78*(5), 337-347.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Neural Networks, 1995. Proceedings., IEEE International Conference on*, *4*.

Kimura, H., Akiyama, S., & Sakurama, K. (1999). Realization of dynamic walking and running of the quadruped using neural oscillator. *Autonomous Robots*, *7*(3), 247-258.

Kimura, H., & Fukuoka, Y. (2004). Biologically inspired adaptive dynamic walking in outdoor environment using a self-contained quadruped robot:'tekken2'. *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, *1*.

Kun, A. L., & Miller III, W. T. (1999). Control of variable speed gaits for a biped robot. *Robotics & Automation Magazine, IEEE*, *6*(3), 19-29.

Matsuoka, K. (1985). Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biological Cybernetics*, *52*(6), 367–376.

Matsuoka, K. (1987). Mechanisms of frequency and pattern control in the neural rhythm generators. *Biological Cybernetics*, *56*(5), 345–353.

Michel, O. (2004). Webots: Professional mobile robot simulation. *Journal of Advanced Robotics Systems*, *1*(1), 39–42.

Mochizuki, G., Sibley, K. M., Esposito, J. G., Camilleri, J. M., & McIlroy, W. E. (2008). Cortical responses associated with the preparation and reaction to full-body perturbations to upright stability. *Clinical Neurophysiology*, *119*(7), 1626–1637.

Mombaur, K., Bock, H., & Longman, R. (2000). Stable, unstable and chaotic motions of bipedal walking robotswithout feedback. *Control of Oscillations and Chaos, 2000.*

Proceedings. 2000 2nd International Conference, *2*.

Mombaur, K., Longman, R., Bock, H., & Schlöder, J. (2005). Open-loop stable running. *Robotica*, *23*(01), 21–33.

Morimoto, J., Endo, G., Nakanishi, J., Hyon, S., Cheng, G., Bentivegna, D., et al. (2006). Modulation of simple sinusoidal patterns by a coupled oscillator model for biped walking. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 1579–1584.

Morimoto, J., Nakanishi, J., Endo, G., Cheng, G., Atkeson, C. G., & Zeglin, G. (2005, April). Poincaré-map-based reinforcement learning for biped walking. *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2381-2386.

Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., & Kawato, M. (2004). Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, *47*(2-3), 79-91.

Park, J. H., & Kwon, O. (2001). Reflex control of biped robot locomotion on a slippery surface. *4*.

Pearson, K. (1993). Common Principles of Motor Control in Vertebrates and Invertebrates. *Annual Reviews in Neuroscience*, *16*(1), 265–297.

Popovic, M. B., Goswami, A., & Herr, H. (2005). Ground reference points in legged locomotion: Definitions, biological trajectories and control implications. *The International Journal of Robotics Research*, *24*(12), 1013.

Riemann, B., & Lephart, S. (2002a). The Sensorimotor System, Part II: The Role of Proprioception in Motor Control and Functional Joint Stability. *Journal of Athletic Training*, *37*(1), 80.

Riemann, B., & Lephart, S. (2002b). The Sensorimotor System, Part I: The Physiologic Basis of Functional Joint Stability. *Journal of Athletic Training*, *37*(1), 71.

Righetti, L., Buchli, J., & Ijspeert, A. (2006). Dynamic Hebbian learning in adaptive frequency oscillators. *Physica D: Nonlinear Phenomena*, *216*(2), 269–281.

Righetti, L., & Ijspeert, A. (2006). Programmable central pattern generators: an application to biped locomotion control. *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 1585–1590.

Righetti, L., & Ijspeert, A. (2008). Pattern generators with sensory feedback for the

control of quadruped locomotion. *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 819–824.

Sardain, P., & Bessonnet, G. (2004). Forces acting on a biped robot. center of pressure-zero moment point. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, *34*(5), 630-637.

Schillings, A., Wezel, B. van, Mulder, T., & Duysens, J. (2000). Muscular Responses and Movement Strategies During Stumbling Over Obstacles. *Journal of Neurophysiology*, *83*(4), 2093–2102.

Sugihara, T., Nakamura, Y., & Inoue, H. (2002). Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control. *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, *2*.

Taga, G. (1998). A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance. *Biological Cybernetics*, *78*(1), 9-17.

Taga, G., Yamaguchi, Y., & Shimizu, H. (1991). Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, *65*(3), 147-159.

Thach, W., Goodkin, H., & Keating, J. (1992). The Cerebellum and the Adaptive Coordination of Movement. *Annual Reviews in Neuroscience*, *15*(1), 403–442.

Vukobratovic, M., & Juricic, D. (1969). Contribution to the synthesis of biped gait. *Biomedical Engineering, IEEE Transactions on*, 1-6.

Yang, J., & Stein, R. (1990). Phase-dependent reflex reversal in human leg muscles during walking. *Journal of Neurophysiology*, *63*(5), 1109–1117.

Zaier, R., & Kanda, S. (2007). Piecewise-Linear Pattern Generator and Reflex System for Humanoid Robots. *Robotics and Automation, 2007 IEEE International Conference on*, 2188–2195.

Zaier, R., & Nagashima, F. (2006, Oct.). Motion pattern generator and reflex system for humanoid robots. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 840-845.

Zehr, E., & Stein, R. (1999). What functions do reflexes serve during human locomotion? *Progress in Neurobiology*, *58*(2), 185–205.

Appendix

A.1 Start of Locomotion Optimization

Resulting wave patterns for the four frequency gaits for the start of locomotion optimization experiment. The lateral ankle patterns are not included since they are in exact anti-phase of the lateral hip joints.

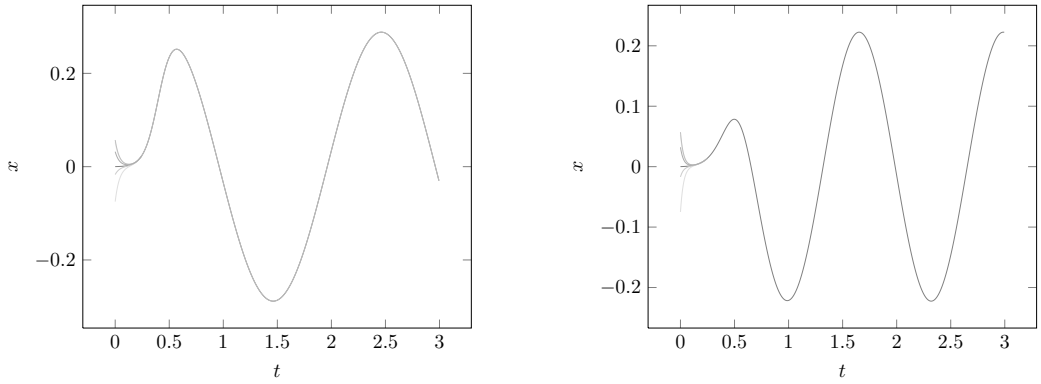

Figure A-1: Lateral Hip (0.50 Hz and 0.75 Hz)

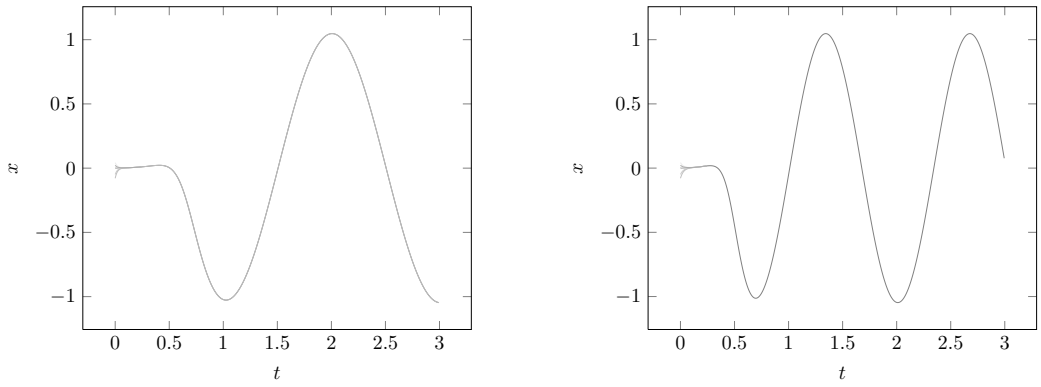

Figure A-2: Frontal Hip (0.50 Hz and 0.75 Hz)

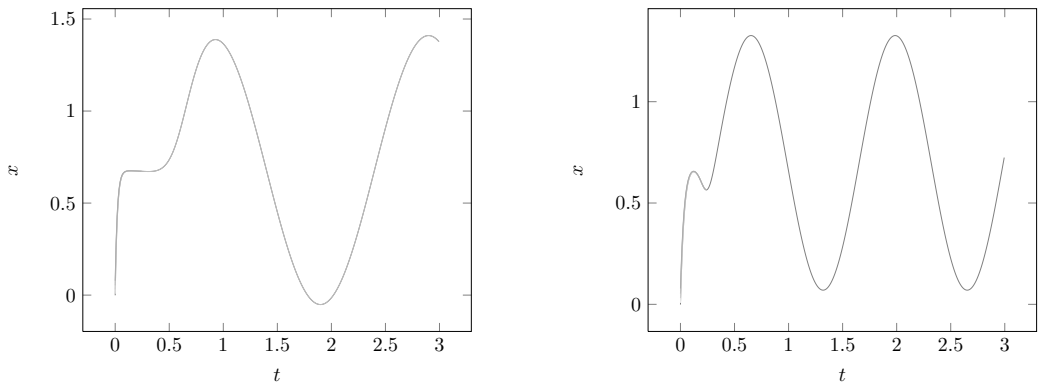

Figure A-3: Knee (0.50 Hz and 0.75 Hz)

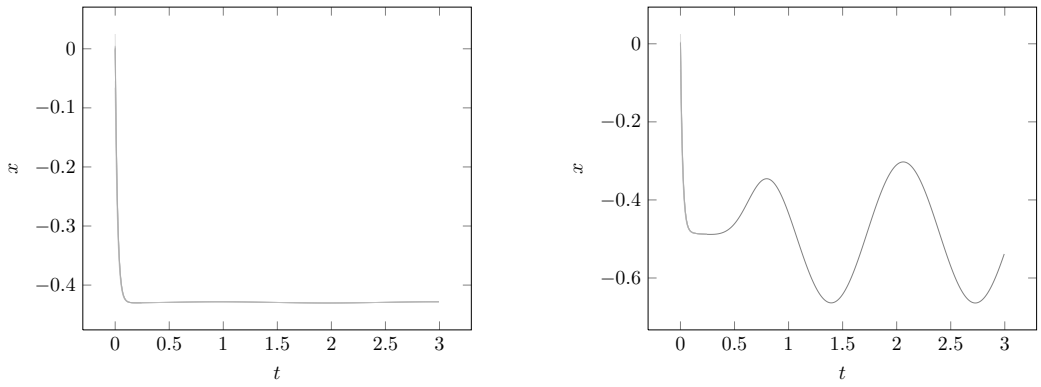

Figure A-4: Frontal Ankle (0.50 Hz and 0.75 Hz)

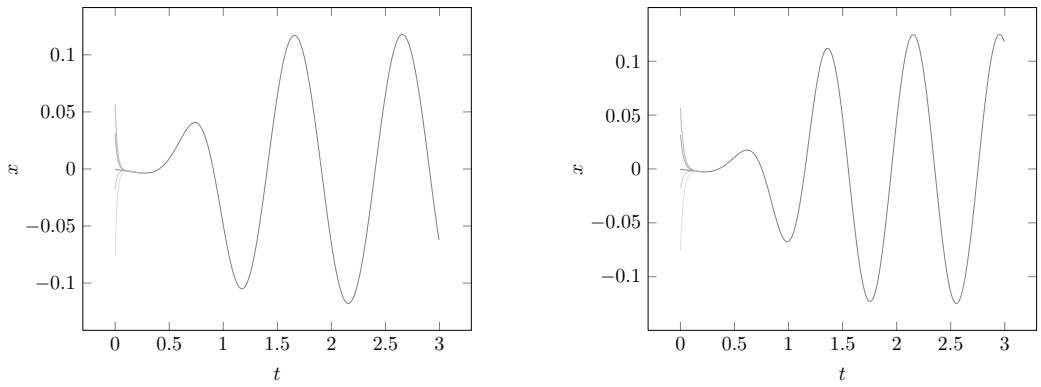

Figure A-5: Lateral Hip (1.00 Hz and 1.25 Hz)

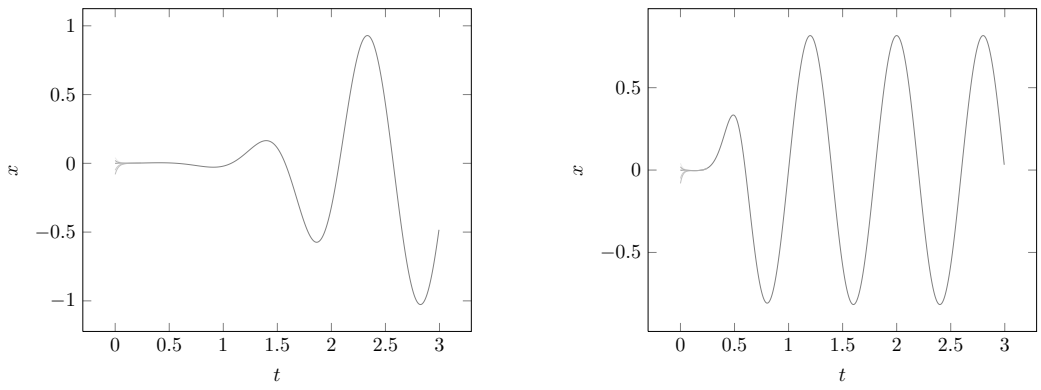

Figure A-6: Frontal Hip (1.00 Hz and 1.25 Hz)

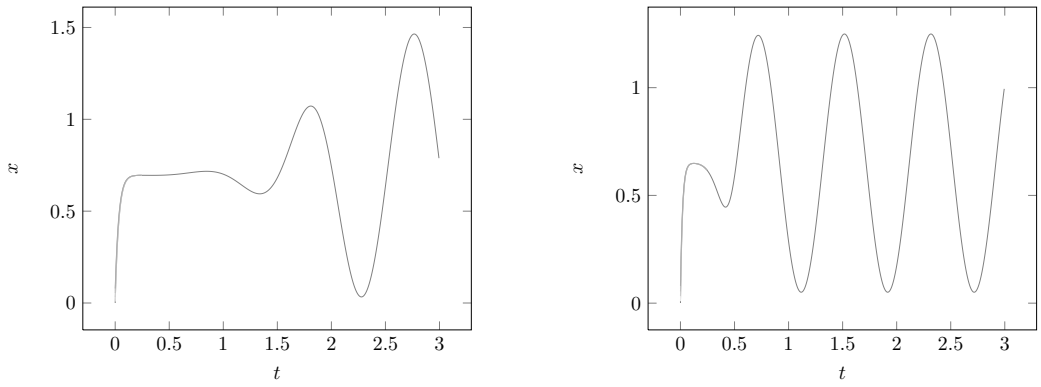

Figure A-7: Knee (1.00 Hz and 1.25 Hz)

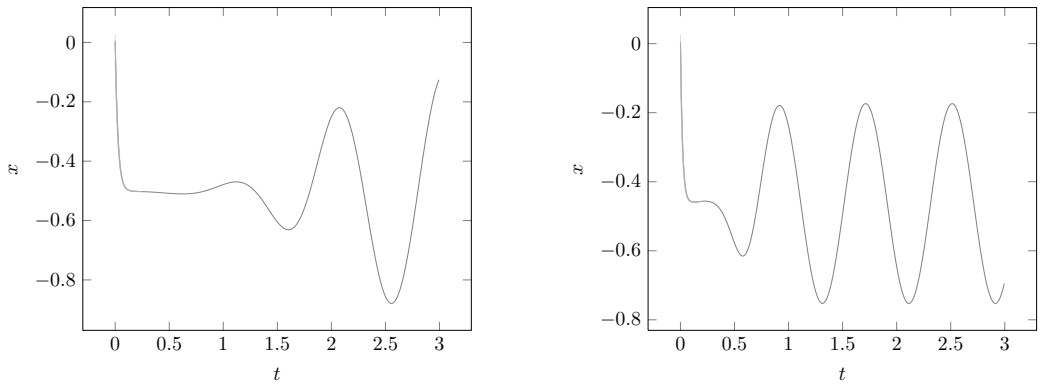

Figure A-8: Frontal Ankle (1.00 Hz and 1.25 Hz)